

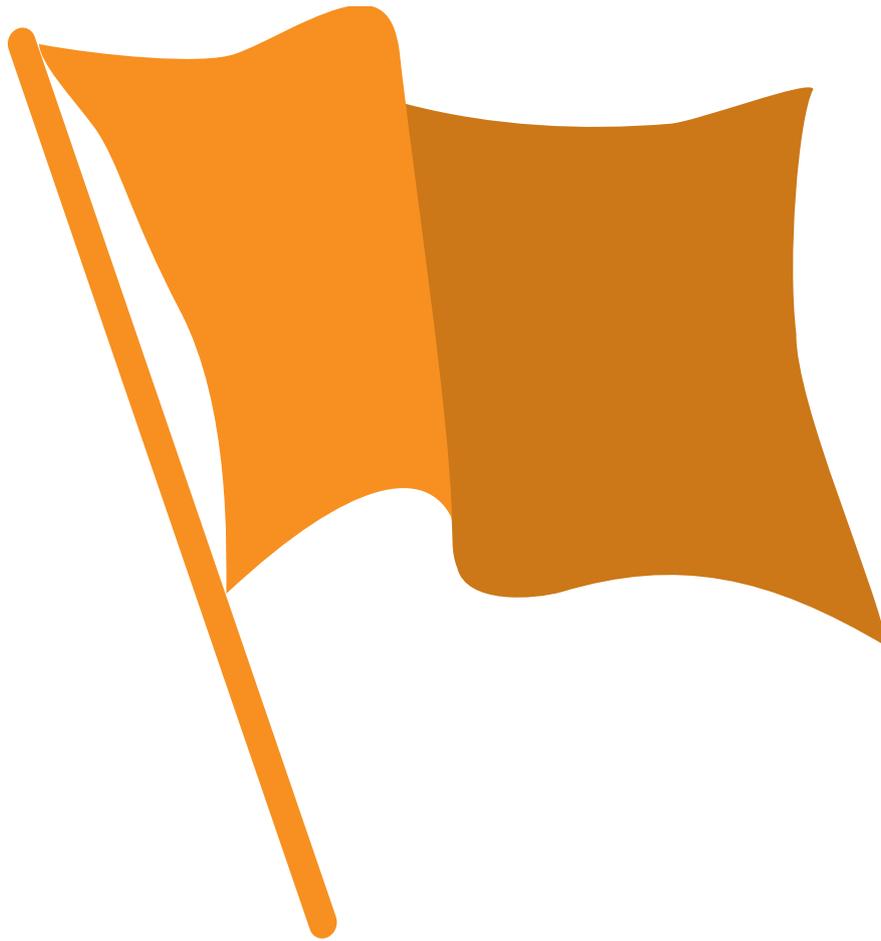


How Better Requirements Management Can Get You Off to a Great Start

Business Management brought to you by

nimblex

One system. Countless possibilities



Introduction

Talking to colleagues and friends about your upcoming projects can be a real eye-opener. Most people who have done this before have heard some war stories on how “everything all went pear shaped”.

No doubt that the successful gathering of Requirements can make or break your new application or process automation project.

What are ‘Requirements’?

Requirements are capabilities that a solution and/or application must provide, or a condition that it must meet, so that ultimately it can solve a problem or help to meet an objective.

Defining a Good Requirement

Because Requirements are the foundation of any application development project, teams need to understand the attributes of a good requirement. The best Requirements are:

1. Actionable
2. Measurable
3. Testable
4. Related directly to business processes/goals -or- Meet the needs of an opportunity that has been identified by the business.
5. Documented with enough detail to ensure that the system can be designed with no ambiguity.

Requirements must form part of a sentence. These sentences must consistently use the verbs “*shall*,” “*will*” or “*must*” to show the requirement’s mandatory nature, and “*should*” or “*may*” to show that the requirement is optional.

The whole requirement specifies a desired end goal or result, and contains a success criterion or other measurable indication of the quality.

The likelihood of an application development project coming in on budget, on time, and satisfying the needs of the stakeholders, is greatly increased when all stakeholders work together towards a shared vision brought about by insight into each other’s perspectives/needs.

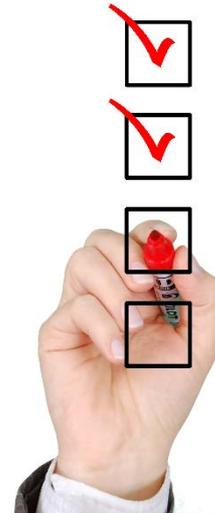
On the other hand, a lack of detailed Requirements can lead to issues and tension between the stakeholders during the project. In the worst-case scenario, it may lead to the development of a system that is not fit for the desired purpose.

As a result, Requirements are an integral part of the initiation stage of a software development project. A robust process at the Requirements definition stage helps to mitigate any ambiguity allowing for a **smoother running and more clearly focused project**.

An important point to keep in mind: Just as it would be unrealistic to embark on any application development project without a set of Requirements, it would be equally as unrealistic to assume that once identified that the Requirements would be a fixed target until the project is delivered. The **Requirements will, in fact, change** as issues arise during the project, or as part of normal business practice evolution.

Types of Requirements

- Business Requirements
 - This typically describes the 'what' and 'why' of a project, e.g. the goals, objectives, future state, etc. They are often expressed by statements like "we need a way to...", or "I wish we could do...".
 - These Requirements solely focus on what needs to happen and does not focus on the software features. It comes from the perspective of the user and the business problem.
 - You can also harvest Business Requirements from Functional Requirements by frequently asking the question "Why".
- Solution / Application Requirements
 - These Requirements are split into two types:
 - Functional Requirements (describes how the application should behave and its capabilities):
 - Focus on what the application must be able to do.
 - Include features and behaviours written from the user's perspective.
 - Are best represented by Use Cases which indicates the desired functionality, needs, and wants, which the user desires in the new application.
 - Example: "The new contract management application must be able to assign a unique contract number to each contract".



- Non-Functional Requirements (describes the system environment and conditions wherein the application must operate):
 - Focus on types of Requirements which don't come from observing users or from what they wish to do with the new application.
 - Include Requirements relating to security, performances, reliability, compliance, etc.
 - Example: "The new contract management application must export data to CSV files".
- Transition Requirements
 - Focus on what is needed to enable moving from a current to a future state.
 - For example, the training of users, data migration, application deployment, etc.

When defining these types of Requirements, keep the following in mind:

- Requirements must be defined so that they are consistent, feasible, and verifiable.
- They must also be written in a manner that does not address how the goal is going to be achieved, but only the goal.
- In addition, it is always important to remember to be explicit. Define every requirement; go through the processes with a fine-tooth comb, and remember that obvious things also need capturing. It should never be assumed that certain functionality will be there. As each stakeholder views the process from one perspective, the functionality that they assume as standard may not be apparent from somebody else's perspective.



Lookout for the Design Trap

Be careful not to step into the design trap. This is a common mistake when we mix specifying what needs to be done and how it is to be done in our Requirements Gathering initiative.

For example, this problem may occur the minute you start, when the user begins to use physical terms, for example, "I'd like to be able to click on a table and". This immediately assumes a certain type of solution. This is an easy mistake because we close our eyes and envision or foresee the physical solution, and then we proceed to describe it to others in physical terms.

Focusing on the 'how' instead of the 'what' can lead to the incorrect description of Requirements and many other issues.

It is a good idea to discuss these things before you embark on this journey with your team and potential users.

We always encourage our clients, while working alongside eBMS, to best focus on the 'what' (logical), i.e. needs and wants. EBMS will then focus on the application design and 'how' to best use all the Nimblex Platform features (physical).

Making Requirements Development Manageable

There are a set of Requirements activities that can be undertaken for every project that will break down the process into manageable pieces.

1. **Requirements Gathering** – The Requirements Gathering stage is about capturing the data needed in order to interpret a set of requirements. There are many techniques available for use during this stage and the most appropriate ones should be chosen in order to identify the following information:
 - full scope of the project
 - who all the stakeholders are
 - what the goals of the project are
 - what tasks the system must perform
2. **Requirements Analysis** – This process involves reviewing all of the data gathered during the Requirements Gathering stage. It is often necessary at this stage to get back to the stakeholders in order to further clarify their needs/positions. Negotiation is often significant at this stage as there will be a certain amount of conflict between different stakeholder's priorities and needs; it is imperative to resolve these prior to moving on.
3. **Requirements Specification** – This is the translation of all the data gathering and interpretation of data into a coherent Requirements Document that identifies all the Business Requirements, User Requirements, Functional Requirements, and Non-Functional Requirements. This document is likely to include any relevant models as well.
4. **Requirements Validation** – This stage requires that all the Requirements identified in the document accurately reflect all the stakeholders negotiated needs, and that the resulting system models are an accurate reflection of the system to be implemented. The knowledge of all the stakeholders is used to verify the document and any resulting issues are documented, clarified, negotiated until all conflicts are eradicated and a complete, consistent and unambiguous Requirements Document is produced.

At first glance Requirements Developments may seem to be a highly complex process, however, they do scale with the size of the project involved. So, when used on a small project, there are likely to be less stakeholders involved and therefore less negotiation and therefore less iterations of the process required.

The Requirements Gathering Process

This process includes the following activities:

- Planning
- Preparing
- Gathering
- Recording / Documenting
- Verifying
- Agreeing

No doubt that the above process needs to be managed, which is what we call 'Requirements Management'.

What makes good Requirements Management?

Definition - Requirement Management

Requirements need to be elicited, analysed, negotiated, specified, validated, tracked, and changed/updated during the life of a project.

All of this has to take place within a controlled environment so that each requirement can be traced back to a specific need and also traced to a function/feature in the system being developed. *"Requirements Management" is the control framework that governs this process.*

Description - "What" and "Why" Behind Requirements Management

It is vital that the Requirements Management function is in place from inception of the project right through to post implementation of the system. One of the goals of this function is to ensure that there are no inconsistencies between the requirements that are identified and the product that is delivered. The process should be robust enough so that if any inconsistencies do arise that they are flagged and rectified in a timely manner.

Requirements Management process must ensure that:

- negotiations between the stakeholders and project team are facilitated
- all Requirements are fully negotiated, defined, and prioritised between the various stakeholders
- a coherent and complete Requirements Document is issued, agreed upon, and kept up to date during the lifecycle of the project
- commitment to the Requirements are given by all stakeholders and the sponsor of the project
- any changes to Requirements during the project lifecycle are reviewed, verified, negotiated, approved and implemented
- all changes are fully tracked and traceable
- all Requirements are documented

Requirements Management keeps the Requirements Documentation up to date and ensures that any changes that are required are reviewed thoroughly prior to approval being given. All changes are tracked for compatibility and to ensure that the project is not creeping off course.

Requirements Management calls for meticulous attention to detail, a holistic and impartial view of the project.

Requirements Management role one of the most important for any software development project. The difference between an effective or an ineffective Requirements Management process **can have a huge influence on the likelihood of success for any given project.**

Who leads the Requirement Management process?

As part of our services, eBMS can facilitate the Requirements Gathering and Management Process for clients, however, most of our clients do this in their own capacity.



If you decide to keep this activity in-house, then the Requirements Gathering and Management Process needs to be led by

- Someone in your organisation who have credibility with the users and their area. Users will be more likely to share their thoughts with someone they trust. This will avoid a 'them and us' attitude which will not be helpful.
- Someone with excellent listening skills and who can communicate with clarity. This will help to analyse, to validate, to double check and to clarify Requirements as received from users.
- Someone who can be firm yet wise with the management of a project plan.

The eBMS team work best if we can have a project owner from the client side who fits the above description. All projects are a partnership between the vendor and client.

Benefits of a Good 'Requirements Management' Process

A good Requirements Management process will provide the following benefits to any organization:

- Lower costs – due to reduction in defects and rework which reduce costs massively.
- Time savings – due to better time efficiency and fewer unnecessary features.
- Higher customer satisfaction levels – due to better communication, no expectations gap and improved quality of the end product.

Common Mistakes in Requirement Management

The reality is that gathering requirements is a lot of work. Project teams can make bad assumptions, focus on the 'how' instead of the 'what' and incorrectly describe Requirements.

A good Requirements Management process is critical in keeping software projects on track. Mistakes in managing Requirements can lead to:

- Project failure
- Project overspends
- Projects running overtime

Some software application projects are not as successful as they could be. Some software development projects are total disasters. The projects that fall into these categories tend to have some or all of the following mistakes in common:

1. **No comprehensive set of Requirements** – It is a poor practice to start developing an application without a set of Requirements. Unfortunately, this is not unheard of. Projects starting based on a conversation or email will undoubtedly lead to an unsatisfying experience for all parties. Mitigate this problem, even on a small project, by following the Requirements management process of:
 - Gathering
 - Analysis
 - Specification
 - Validation
 - Management
2. **Ambiguous Requirements** – Requirements must not be open to different interpretations by the stakeholders. This is a difficult mistake to avoid but it can be mitigated. The validation process is where any ambiguity should be identified. When the draft Requirements Document is issued, it should include test cases and system requirements. This will ensure that the likelihood of the document being interpreted differently by different stakeholders is minimized.

3. **Not getting all the stakeholders involved in the Requirements Process from the beginning**

– It is very important to get all stakeholders involved to avoid Requirements falling through the gaps. Adding new Requirements that could have been identified at the projects outset can cause huge issues for the project,



ranging from lengthened project times to increased cost. This mistake is easily mitigated. When a project is initiated the business process should be reviewed from end to end. Carrying out this review will ensure that every stakeholder is identified. Once identified they should be fully engaged with throughout the lifecycle of the project.

4. **Not adequately managing the Requirements after the initiation stage** – Firm management and control are required to ensure that there is minimal Requirements creep. A couple of things that will help mitigate this are:

- Clearly identifying the scope of the project from the outset. This will help deflect new Requirements that fall outside the original scope.
- Have a robust process for reviewing, validating and approving change requests.

5. **Developers including functionality that is not in the Requirements Document** – Developers can sometimes get a little carried away, so they have to be managed in the same way as the other stakeholders. Sometimes developers can be overtaken by creativity and try to add unrequested functionality. This is known as “gold plating”. This can be mitigated by making sure that the developers keep to the “KISS” principle. ‘Keeping it simple’ means creating a robust, lean and simple design. This mistake is mitigated by ensuring that traceability of the functionality is bidirectional to the Requirements.

The biggest cause of overspend and overrun in software projects is rework cause by the mistakes outlined above and related mistakes. Requirements that are properly elicited, defined, validated and managed will help maximize the overall success of a project.

Conclusion

Requirements definition and management are among the most important activities in any project, and efforts in this direction can improve and accelerate ROI. It is also the first area to focus on, based on the “garbage in, garbage out” rule: If the Requirements are not clear, any other effort may just help you produce the wrong application faster.

The first step to better Requirements management is to understand the simple rules that make a requirement “good.”

EBMS can help to manage Requirements more effectively, but we can also help with the design and application configuration on the Nimblex Platform.

If you would like to know more about our Nimblex Platform, check out our website: getnimblex.com, drop us an email: <mailto:info@ebms.com.au>, or give us a call on 1300 721 159.