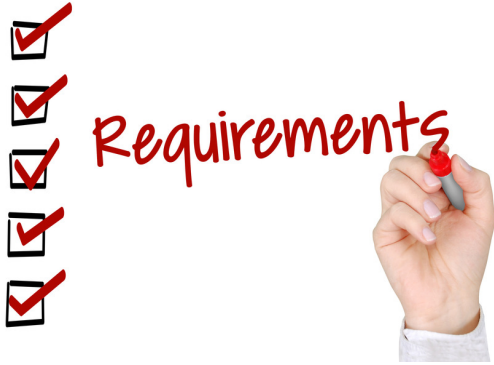


## Defining Requirements



The scope of any project should represent the minimum amount of work needed to fulfil the customer's, or end user's requirements. Doing more work is wasteful; doing less means the client's requirements are not fulfilled. Therefore, to get the scope right you have to get the requirements right first! And that means asking the right people the right questions.

Optimising the scope of work of the project, to deliver maximum value requires the following steps:

- Ensure the objectives the project is being created to fulfil are clearly defined. The objectives should be directly linked to the creation of value for the organisation<sup>1</sup>.
- Identify the key stakeholders<sup>2</sup> who are likely to have specific requirements that require fulfilling.
- In conjunction with the key stakeholders, define all of the requirements needed to achieve the objectives. Both the objectives and the requirements may be more than simply the deliverables handed on to the customer; eg, the organisations governance systems may require reports to fulfil the organisations objective of good governance. A range of data gathering techniques should be used to ensure all of the requirements are gathered and understood (see below).
- If necessary, optimise the selection of the requirements<sup>3</sup> to be included in the final scope to balance value, time, cost and quality objectives.
- Record the requirements the project will achieve in an appropriate system to support the development of the project's scope baseline<sup>4</sup>.

Requirements should not be viewed in isolation, but rather viewed from the perspective of how they relate to the whole; they are the common thread that ties all system development activities and work products together. Consequently, writing requirements should be viewed as an exercise in systems<sup>5</sup> engineering, not an exercise in documentation.

### Factors to consider in developing a set of requirements

The following elements should be considered to make sure all of the requirements are fully understood:

- 
- <sup>1</sup> For more on **benefits and the 'value chain'** see: [https://www.mosaicprojects.com.au/WhitePapers/WP1023\\_Benefits\\_and\\_Value.pdf](https://www.mosaicprojects.com.au/WhitePapers/WP1023_Benefits_and_Value.pdf) and [https://www.mosaicprojects.com.au/WhitePapers/WP1042\\_Outputs\\_Outcomes\\_Benefits.pdf](https://www.mosaicprojects.com.au/WhitePapers/WP1042_Outputs_Outcomes_Benefits.pdf)
  - <sup>2</sup> For more on **stakeholders** see: [https://www.mosaicprojects.com.au/WhitePapers/WP1007\\_Stakeholder\\_Cycle.pdf](https://www.mosaicprojects.com.au/WhitePapers/WP1007_Stakeholder_Cycle.pdf)
  - <sup>3</sup> For more on **ranking requirements** see: [https://www.mosaicprojects.com.au/WhitePapers/WP1062\\_Ranking-Requirements.pdf](https://www.mosaicprojects.com.au/WhitePapers/WP1062_Ranking-Requirements.pdf)
  - <sup>4</sup> For more on the **Requirements Traceability Matrix** see: [https://www.mosaicprojects.com.au/WhitePapers/WP1029\\_Requirements\\_Traceability\\_Matrix.pdf](https://www.mosaicprojects.com.au/WhitePapers/WP1029_Requirements_Traceability_Matrix.pdf)
  - <sup>5</sup> For more on **systems thinking** see: [https://www.mosaicprojects.com.au/WhitePapers/WP1044\\_Systems\\_Thinking.pdf](https://www.mosaicprojects.com.au/WhitePapers/WP1044_Systems_Thinking.pdf)



- Ask potential users of the projects deliverables what they need. Consider each potential user group separately as their needs will be different.
- Define and prioritise the required technical functions for each deliverable.
- Be sure that organisations process is mature and stable, and then be sure that the deliverables can support the processes. This may suggest a less sophisticated deliverable, or alternatively identify additional requirements for process improvement and training<sup>6</sup>.
- Consider how the deliverables should integrate with other tools and processes used by the organisation and the wider business.
- Define how the deliverables should work once completed and the degree of standardisation or flexibility needed both in the short term and the longer term.
- Consider what training the users might need to be able to use the deliverables properly. Who will be responsible for the training needs analysis, developing the training materials, training the trainer and training the users?
- Decide on scalability and the ability to manage the full spectrum of current needs and likely longer-term needs. Should the deliverable be a ‘one-size-fits-all’ solution or focused on a particular type or size of need/problem/requirement. The cost and complexity of the project will vary significantly.
- Think about what ongoing support the organisation might need from external suppliers and vendors (or from your organisation if the client is external).
- Assess how the chosen solution might need to grow with your business.
- Determine the budget, including purchase/development of the deliverable, possible customisation, piloting, data cleansing/migration, training, communication and roll-out. But cost should not be the driving consideration - you will get what you pay for and you should buy what you need based on the best value proposition.

These factors can be compiled into a *functional requirement specification* defining what is needed from the deliverable. This categorises requirements into those which are essential, the ones that are preferred, and optional extras.

If purchasing from an external source, you can then use this specification to screen available options and produce a short-list of possible candidates that meet all or most of the criteria. Invite the vendors of these solutions to present their ‘tool’ in more detail. Try testing each ‘tool’ using actual data to ensure that the reality lives up to the vendor’s sales pitch. Invite real users to take part in trials to give you their feedback on whether it meets their needs. See if vendors are able to tailor their ‘tool’ to meet your specific requirements.

### Quality metrics for requirements

The quality of the requirements you gather is critical for the overall success of a project or program. They should be assessed against the following criteria to determine whether requirements are:

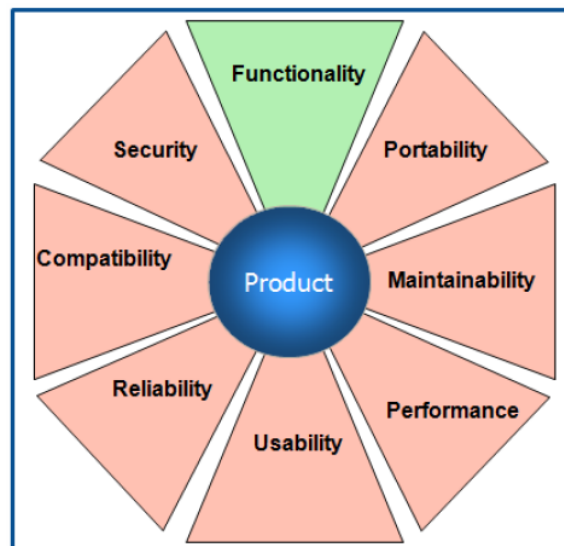
---

<sup>6</sup> For more on **process improvement** see:  
[https://www.mosaicprojects.com.au/WhitePapers/WP1046\\_Process\\_Improvement.pdf](https://www.mosaicprojects.com.au/WhitePapers/WP1046_Process_Improvement.pdf)



- **Unambiguous:** Every statement is assessed to ensure that it has one interpretation only and that terms are clear and well defined. Language used in the statement must not leave a doubt in the reader's mind as to the intended descriptive or numeric value;
- **Complete:** The stated requirement is complete and does not need further amplification;
- **Verifiable:** The statements should not be vague or general but quantified in a manner that can be verified;
- **Consistent:** Statements are assessed to ensure that conflicting terminology, contradictory required actions, and impossible combinations are absent. It should also include an assessment that the set of requirements does not have individual requirements, which are contradictory. Requirements should also not be duplicated. The same term should be used for the same item in all requirements;
- **Traceable:** Each referenced statement is assessed to ensure it is uniquely identified, via a project unique identifier. Traceability between artefacts of each development phase will be maintained;
- **Testable:** Can the requirement be tested during development to know that it has been delivered;
- **Verified:** Agreed by the relevant stakeholders with authorisation to proceed with its implementation; and
- **Correct:** The stated requirement is said to be 'correct' if it represents something required of the system to be built. There should be no requirements that relate to other systems (i.e.: elements of other systems, outside the scope of the current specification).

Care needs to be taken to ensure the requirements also contain all of the necessary non-functional requirements (NFRs). End users don't necessarily detail (and are frequently unaware of) NFRs and requirements gatherers don't always ask the right questions<sup>7</sup> to properly understand what quality characteristics the project's outputs should satisfy.



Typical NFR requirements

---

<sup>7</sup> For more on *asking questions* see: [https://www.mosaicprojects.com.au/WhitePapers/WP1012\\_Active\\_Listening.pdf](https://www.mosaicprojects.com.au/WhitePapers/WP1012_Active_Listening.pdf)

NFRs related areas such as security, performance, usability, reliability etc (also referred to as structural components). The challenge in eliciting NFRs is that they are generally not stated explicitly and are not as clear in stakeholders' minds as functional requirements. If the product quality requirements are not clearly stated, it will be very difficult to ensure they are achieved in the design and development phases of the project.

### Skills required for effective requirements analysts

Some traits of effective analysts, needed to identify the requirements of multiple internal customers, and particularly relevant to the work of reconciling their legitimately conflicting interests, are as follows.

- The analysts must have a strong ability to deal with customers and extract from them a sense of what they truly need
- They must have good political skills and recognise that all customers (stakeholders) are not equal in a political sense.
- They must be technically competent and able to match customers' ill defined needs to possible solutions
- They must be open-minded and possess a good imagination. Open-mindedness is particularly important so the analyst does not close off possible solutions to problems because of a narrow outlook
- They must have a high tolerance for ambiguity because customers do not generally know what they need or want, they tend to provide the needs analyst with mixed signals.
- They must be articulate.

### Some useful techniques

Knowing your audience will help you determine the right data gathering<sup>8</sup> techniques to use to get the most relative information.

1. **One-on-one interviews.** The most common technique for gathering requirements is to sit down with the clients and ask them what they need. The general flow of the interview and key questions should be planned in advance starting with 'open questions' before moving to closed questions focused on specific items. Be ready to ask difficult questions<sup>9</sup>!
2. **Group interviews.** These are similar to the one-on-one interview except that there is more than one person being interviewed and more planning and structure is needed. Skilled questioning helps get the information you need quicker than one-on-one interviews and encourages creative interaction between all of the participants.
3. **Facilitated sessions.** In a facilitated session, you bring a larger group together for a common purpose. In this case, you are trying to gather a set of common requirements from the group in a

---

<sup>8</sup> For more on **data gathering** see: [https://www.mosaicprojects.com.au/WhitePapers/WP1068\\_Data\\_Gathering.pdf](https://www.mosaicprojects.com.au/WhitePapers/WP1068_Data_Gathering.pdf)

<sup>9</sup> For more on **asking the right questions** see: [https://www.mosaicprojects.com.au/WhitePapers/WP1012\\_Active\\_Listening.pdf](https://www.mosaicprojects.com.au/WhitePapers/WP1012_Active_Listening.pdf)



faster manner than if you were to interview each of them separately then have to reconcile differences. Facilitation is a skilled process<sup>10</sup>.

4. **JAD sessions.** Joint Application Development (JAD) sessions are similar to general facilitated sessions. However, the group typically stays in the session until a complete set of requirements is documented and agree to.
5. **Questionnaires.** Are good tools to gather requirements from stakeholders in remote locations or those that will have only minor input into the overall requirements. Questionnaires can also be the only practical way to gather requirements from large numbers of people.
6. **Prototyping.** Build an initial version of the solution – or part of the solution. You show this to the client, who then gives you feedback and additional requirements.
7. **Following people around.** This is especially helpful when gathering information on current processes. You may need to watch people perform their job before you can understand the entire picture. In some cases, you might also like to participate in the actual work process to get a hands-on feel for how the business function works today.

## The Requirements Management Plan

The Requirements Management Plan describes how you will elicit, analyse, document and manage the requirements of the project, including:

- The initial gathering of high-level project and product requirements early in the planning process;
- The gathering of more detailed product requirements collected during the project lifecycle;
- The approval of requirements by key stakeholders;
- The management of changes to the requirements after they have been approved.

The plan should include the following information:

- **The requirements gathering process.** In this section you will describe the process that you will use to elicit, analyse and document the requirements (the focus of this White Paper).
- **Roles and responsibilities.** Who will be involved with managing the requirements through the rest of the project lifecycle and their responsibilities. Roles could include the project manager, lead analyst, clients, etc. The project manager, for instance, should have the overall responsibility for scope change management of the requirements. Someone, perhaps the lead analyst, should have overall responsibility for the integrity of the requirements throughout the rest of the lifecycle.
- **Tools.** Describe any (automated) tools that will be used to document, manage and track requirements throughout the lifecycle.
- **Requirements traceability<sup>11</sup>.** The overall process should be described here, with key steps added to the schedule to ensure the proper tracking of requirements occurs throughout the rest of the project.

---

<sup>10</sup> For more on **facilitation** see: [https://www.mosaicprojects.com.au/WhitePapers/WP1067\\_Facilitation.pdf](https://www.mosaicprojects.com.au/WhitePapers/WP1067_Facilitation.pdf)

<sup>11</sup> For more on the **Requirements Traceability Matrix** see: [https://www.mosaicprojects.com.au/WhitePapers/WP1029\\_Requirements\\_Traceability\\_Matrix.pdf](https://www.mosaicprojects.com.au/WhitePapers/WP1029_Requirements_Traceability_Matrix.pdf)



- **Change control.** There should be a formal process to manage changes to the requirements. Hopefully, the entire project is using a formal scope change process. If so, then this overall scope change process should be specifically applied to the changes in requirements. If there is no formal overall scope change process, a specific change control process should be documented here.

This plan forms part of the overall project management plan, and adhering to a formal requirements management process helps the project team focus on the requirements that have been developed in response to stakeholder inputs, maintains the integrity of the requirements throughout the lifecycle of the project, and contributes to meeting the stakeholder's expectations of a successful project outcome.

### The problem with requirements!

Business people should not be asked to sign off on a set of requirements they cannot understand. No one can read 3,000 requirements statements and be confident they represent the business outcomes that are desired. Requirements are technical statements; the business needs to understand what these precise technical bits and pieces will actually achieve or deliver in terms of capability and functionality.

To achieve this they need a model that clearly shows in their terms how they will work in the future using the application or processes described by the 'requirements'. The model needs to include detail of processes, interaction with systems, roles and responsibilities and enough detail so business people can clearly see in concrete terms how they will operate in the future. A similar model needs to be constructed that clearly shows how the same elements of the business operate today.

These current state and future models pay for themselves:

- There are far too many cases where requirements were missed because the detail of current operations was overlooked in the future state model.
- They can be used to quantify the expected benefits by comparing costs and effort between the new and the old, and to test the business case for change<sup>12</sup>.
- They become an operational model that helps the business to operate more consistently and more compliantly.
- They allow better decisions and adjustments to be made during the project to steer towards a successful outcome.

In the worst case, they may suggest the transformation will not provide sufficient returns for the costs and risks identified, and avoid further investment that should not be made. In this event, the current state model can quickly be used as an operating model, which yields significant benefits.

In the best case, the future state model will be used as the new operating model, helping to embed change so the business does not revert to the old, less effective or less compliant way of working.

---

<sup>12</sup> For more on **benefits and value** see:  
[https://www.mosaicprojects.com.au/WhitePapers/WP1023\\_Benefits\\_and\\_Value.pdf](https://www.mosaicprojects.com.au/WhitePapers/WP1023_Benefits_and_Value.pdf)



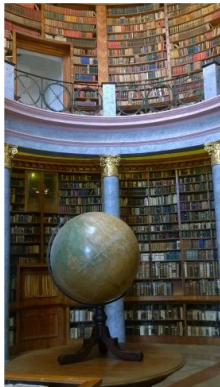
## Summary

Unless the project team and its key stakeholders understand and agree on precisely what has to be delivered by the project to meet its clients and stakeholders needs, the project is very unlikely to be successful.

Even when there is a defined contract or purchase order, making sure all of the requirements are clearly defined and understood is critical to ensuring success. As understanding of the requirements develop, the consequence may simply be that the project team has a better appreciation of what is required to be successful, or alternatively changes may be identified that need managing through the change management process. In either event, the sooner understanding and agreement is achieved, the better for both the stakeholders and the project team.

---

First published 14<sup>th</sup> August 2011 – augmented and updated.



**Downloaded from Mosaic's PMKI  
Free Library.**

For more papers focused on **Scope Management** see:  
<https://mosaicprojects.com.au/PMKI-PBK-015.php>

Or visit our PMKI home page at:  
<https://mosaicprojects.com.au/PMKI.php>



Creative Commons Attribution 3.0 Unported License.

