5 Common Requirements Authoring Mistakes & How to Avoid Them

Writing project requirements, especially for complex or safety-critical products, is a tedious process. Errors are easily introduced at any stage of the process - even before pen meets paper to start writing the requirements document. Here are five ways you may not have thought of that can cause those errors.

QRA

INTRODUCTION

Writing project requirements, especially for complex or safety-critical systems, is a tedious process. Errors are easily introduced at any stage of the process – even before pen meets paper to start writing the document itself. Here are five ways you may not have thought of that can cause those errors.

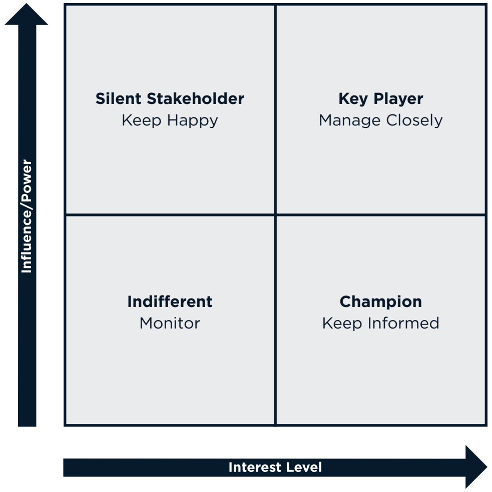
1. MISUNDERSTANDING WHO THE TRUE STAKEHOLDERS ARE

A client calls you or an internal customer sends you an email. This person becomes your primary point of contact and in your mind, a key stakeholder. But that is not always the case. Perhaps this person is a manager who made the initial contact but does not know much detail about what needs to be done. Perhaps a project manager in one department is working with you but there are key decision-makers in another department. There are frequently lower-level product users, even shop floor operators, who are experts in their areas and whose opinions carry great weight with management.

One of the first things a successful project manager does, even before gathering any requirements, is stakeholder mapping. There are many ways to do stakeholder mapping. The simplest is to sit down with your project team to brainstorm a list of everyone who should be involved and who has influence. Be sure you do not overlook anyone. Sometimes the most junior employees can provide the best input, especially around efficiency issues.

A good way to be sure you've captured everyone is to choose several key people and send the entire list of stakeholders to them with the question, “Are these the right people to provide input to this project?”. This will help to uncover any key people or functions that may be missing from the team. It is not uncommon for management to assign people to a project quickly and without considering who the right people really are.

After listing the stakeholders, assign each of them a rating for **interest level** and for **influence/power.** They can be placed on a grid similar to the one shown below, or especially for larger lists can be maintained in a spreadsheet. Develop a plan to keep the stakeholders satisfied, or if necessary to move them to a different place on the grid.



Silent Stakeholder Keep Happy	**Key Player** Manage Closely
Indifferent Monitor	**Champion** Keep Informed

The stakeholder list and stakeholder analysis are live documents that are updated regularly as the project progresses and new information is received.

During this process, build trust with your stakeholders. Give them confidence that you know their business and their needs by taking the time to understand their industry and understand their goals.

2. NOT BUILDING THE REQUIREMENTS IN A COLLABORATIVE WAY

Collaboration is not easy. Even if you want to be collaborative, do the stakeholders? They may just want you to go away and build it because they do not have time to engage – but will then criticize the final product. Most likely, you will need to continually pressure stakeholders to be engaged. You can, and should, make thoughtful suggestions, but how can you build what they want if they do not tell you what they want? In addition, if you have done a thorough job identifying stakeholders, you will likely see competing interests and needs. This is especially common in matrix organizations.

As you hold meetings to gather requirements be sure to run those meetings well. Watch out for people who jump to solutions before understanding the problem, focus on the wrong problem, or jump ahead to design instead of first defining requirements. Be alert to the fact that some people are likely to have dominant personalities and some are likely to keep quiet even when they have a contribution to make. Give everyone a chance to be heard and get their ideas on the table. Be sure to keep the meetings on track. It is easy to lose focus. When disorganized meetings become the norm, people will stop showing up and deprive you of their ideas.

Always a good practice, but especially helpful in keeping track of requirements from multiple sources, is a requirements traceability matrix. Who has the authority to decide? Who made the decision? Why did they make it? If a requirement is deleted, why was it deleted and who requested it? Why was it there in the first place? This can help keep competing interests from overriding each other and allow these issues to be resolved early and in a collaborative way.

It can be tempting to take the output of various meetings, phone calls, and emails and quickly turn them into a final requirements document. However, the better approach is to take that draft document and circulate it among the appropriate team members and stakeholders for agreement prior to finalizing it. This allows everyone to get an overall view of the requirements and how someone else's requirements interact with theirs. If nothing else, getting final and formal agreement protects the project team from later criticism about a lack of collaboration. A word of caution – give everyone time to review the document and make sure they understand it. Hastily moving through the requirements stage and quickly getting final approval for the requirements only to "check the box" will likely lead to stakeholders who do not truly understand what they are getting.

Perhaps the most important aspect of collaboration is communication. Good communication helps surface issues early so they can be resolved. Even when there is nothing to communicate, say so! Otherwise, your stakeholders will assume nothing is happening and no progress is being made.

During this process, build trust with your stakeholders. Give them confidence that you know their business and their needs by taking the time to understand their industry and understand their goals.

3. NOT REALIZING WRITING REQUIREMENTS IS AN ITERATIVE PROCESS

If the project team does not realize requirements authoring is an iterative process, they are likely to move ahead with one of the first sets of requirements that they generate. Then, they get too far down the wrong path before the stakeholders bring the team back to what they really want. Stakeholders feel left out and disengaged and will feel the project team has an arrogance in their approach. There will be a lack of collaboration and lack of buy-in, and going forward the stakeholders will refuse to speak up since they believe the project team will not listen anyway. In the end, you will have built something different from what they really wanted.

Do not start design before understanding the requirements. Jumping to solutions without first analyzing the problem and collaboratively brainstorming the best solutions limits the team to one solution before they know it is the best one and blinds them to the problem they are trying to solve. Never determine the solution before determining the problem.

For many projects, a great way to begin is to map the business process. How does the work flow? Change is an opportunity for improvement, so look for inefficiencies in the current process and use this as an opportunity to remove inefficiencies from the process.

During this process, build trust with your stakeholders. Give them confidence that you know their business and their needs by taking the time to understand their industry and understand their goals.

4. LACK OF CLARITY IN THE REQUIREMENTS

So you have done things right. You have identified all of your stakeholders. You have built your requirements collaboratively and iteratively. But are the resulting requirements clear?

Some important things to keep in mind as you write your requirements are:

Requirements language must never be ambiguous. Avoid weak, vague, or subjective language. Beware of requirements containing the word “should”. Use “shall” and “must” instead. Check for language that implies that a requirement is optional. Requirements are just that – required. Finally, purge your requirements of superfluous and non-descriptive or over-descriptive language. Short requirements are easier to understand.

Use a system. Using a best-practice system such as EARS when writing requirements will help ensure clarity, as will writing in the active voice rather than passive voice.

Avoid mixing functional and non-functional requirements. Keeping functional and non-functional requirements separate will help everyone understand the overall goal of the project. To simplify the non-functional requirements consider writing them broadly. If every page on a website must load in the same amount of time, there is no reason to repeat that non-functional requirement for each page. Read about best practices for functional and non-functional requirements here.

Write testable requirements. If a requirement cannot be measured, you will have no way to know if it has been achieved.

Include a glossary or definitions section. A good requirements document will have a definitions section or a glossary. Every industry and even every company has its own acronyms and terms. Large corporations may even use the same term in different ways in different parts of the business. To a computer programmer, “AI” is artificial intelligence. To a chemist, it is an active ingredient. Without a fixed definition, developers are likely to assume your company defines terms the same way the previous company he worked for did. Avoid ambiguity by defining every term and acronym used in the requirements document.

Be consistent. Clear requirements are consistent in the use of units and terms. If possible, choose a unit system and specific units to use throughout the requirements. Always use terms only as they are defined and never use synonyms.

Use the right document. Exercise caution if you are using a requirements template. If you force requirements for your current project into a standard template or into a requirements document from a previous project, it may not be a fit and can lead to confusion.

To automate your authoring processes and ensure requirements are clear and concise, use QVscribe. This is a tool that allows engineers to quickly analyze requirements documents for vulnerabilities. QVscribe uses Natural Language Processing that proactively checks for compliance with best practices such as INCOSE and tags requirements that may be confusing or unclear. Consistent EARS structure is enforced and many of the common errors above are marked automatically for potential correction.

5. NEVER FREEZING THE REQUIREMENTS

A project whose requirements are never frozen can never be finished. Your developers need to know precisely what you want them to build, otherwise, they will get bogged down in never-ending changes, rework, and scope creep.

Although agile projects are becoming very common, realize that agile does not mean there are not functional requirements or an overarching plan and strategy. Although there can be more room for requirements changes within an agile project than within a traditional project, most agile projects should have their requirements frozen within each sprint. It can be tempting to label a project without coherent requirements “agile”, but agile projects are really a series of small, well-managed projects.

CONCLUSION

Authoring requirements is not an easy task but getting the requirements right will save both time and money in the long run – and improve the quality of the finished product. If you can avoid these five mistakes when generating the requirements for your next project, you will be well on your way to a successful result.

version 1

To learn more about QVscribe, visit qracorp.com/qvscribe

qracorp.com

REFERENCES

“10 Typical Mistakes in Specs.” Yegor’s Blog About Computers, 10 Nov. 2015, https://www.yegor256.com/2015/11/10/ten-mistakes-in-specs.html.

“29148-2018 – ISO/IEC/IEEE International Standard – Systems and Software Engineering — Life Cycle Processes — Requirements Engineering.” IEEE, https://standards.ieee.org/standard/29148-2018.html.

“Agile Requirements Change Management.” Agile Modeling, http://agilemodeling.com/essays/changeManagement.html.

“Essential Guide to Business Process Mapping.” Smartsheet, https://www.smartsheet.com/essential-guide-business-process-mapping.

“Requirements Traceability Matrix – RTM.” Project, 16 Oct. 2018, https://project-management.com/requirements-traceability-matrix-rtm/.

“Stakeholder Analysis.” Wikipedia, Wikimedia Foundation, 1 Oct. 2019, https://en.wikipedia.org/wiki/Stakeholder_analysis.

“The Top 8 Mistakes in Requirements Elicitation.” BA Times, https://www.batimes.com/articles/the-top-8-mistakes-in-requirements-elicitation.html.

Wiegers, Karl Eugene, and Joy Beatty. Software Requirements. Microsoft Press, 2015.