Use open questions and context for a better RFP

7:46 minute read based on [183 words per minute](https://www.sciencedirect.com/science/article/abs/pii/S0749596X19300786?via%3Dihub).

**Table of Content**

[Introduction](#_Toc113890757)

[Open questions give more insights](#_Toc113890758)

[Request for a Partner](#_Toc113890759)

[Provide more context](#_Toc113890760)

[The challenge of comparing](#_Toc113890761)

[Using a scoring matrix](#_Toc113890762)

[Relative scoring](#_Toc113890763)

[Examples of open questions](#_Toc113890764)

[(Data)Privacy](#_Toc113890765)

[Security](#_Toc113890766)

[Architecture](#_Toc113890767)

[Integration](#_Toc113890768)

[Service Levels](#_Toc113890769)

[Implementation](#_Toc113890770)

# Introduction

In the last couple of months, I were involved in many Requests for Proposal. As an architect, I am often asked to provide a list of Non-Functional Requirements to ensure the supplier and potential solution agrees with internal regulations for security, use of data, cloud ambition, etc.

I strongly believe in asking open questions to software and implementation partners as it will give more insight into the potential solution. And by providing more context on the business ambition and architectural vision and principles will turn the RFP from a Request for a Solution into a Request for a Partner to achieve that ambition.

# Open questions give more insights

When I am involved in Requests for Proposal, I start by asking for available non-functional requirements. And most of the time I receive Excel lists with not questions but statements that a partner can ‘check’.

But these lists, and the response by partners, do not provide you with insights on how they achieve this ‘requirement’. Not to mention the level of ‘blue eyes trust’ you must have with the partner’s response. For example, the requirements “The API’s must be documented” or “Data transfer must be secure”, and the partner marks this as ‘compliant’, but it is still unclear how the partner achieves this. For API documentation they might use internal code comments, and for secure data transfer, the partner might believe that SOAP over HTTPS might be sufficient.

Therefore I believe it is crucial to rephrase the non-functional requirements into open questions. Not only will you know if the partner CAN comply, but you will also understand HOW they comply, and therefore can better understand if the level of compliance is good enough.

It will also give you choices as the partner can respond by providing a few options on how to comply. For example, requirements related to the availability of service levels will often result in options you can choose from. Instead of requiring a “99.99% availability”, ask what availability the partner can provide, and they will provide you with options, with consequences for the price of course. When the scope for the RFP is a crucial application that must be always available, you can choose the highest availability that the partner can offer at a high price. But for non-crucial applications, you can choose a lower availability at a lower price.

I have put a list of open questions that I used in RFPs later in this article.

# Request for a Partner

The advantages of open questions to non-functional requirements also apply to business requirements, but the reasoning is different. Too often, I see lists of business requirements created from the NOW. “We work with this application, and it can’t do this, so we need an application that can do this.” But in this rapidly changing world of technology and customer needs, you can never predict what you need shortly. So good changes are you end up with a solution wherein 1-2 years a feature is missing.

“If I had asked people what they wanted, they would have said faster horses.”   
- Henry Ford

I strongly believe in partnership. Most software partners spent millions of euros on product development, much more than most companies can spend. My advice is to use their investment too and build a partnership to achieve your ambition. When you are partners with a software partner and you have an influence on, or at least knowledge of, their product roadmap, this partnership is beneficial and fruitful for both partners. And you are much better prepared for the future, including new technologies and responding to future customer needs.

# Provide more context

To make sure you select the right partner to realize your ambition it is crucial to explain in the RFP your business ambitions. One of the strong points of TOGAF, a software architecture framework, is starting with a business vision and strategy before defining solution architecture.

In the most recent RFPs I was involved I created a context document, attached to the RFP that summarized the business ambition. Followed by a question to partners about how they can help us achieve this ambition with their proposed solution. That was chapter 1. The next chapter explained the architectural vision and ambition for the company’s IT landscape, followed by architectural principles we conducted from that vision. And after each principle – we usually have around ten principles – we ask the partner how our principle is represented in their solution.

**Appendix**

1. Business Vision

2. IT Vision & Strategy

3. Architecture principles

4. Data & Integration principles

5. Security principles

The feedback that I got from software partners so far is that they like this approach. It provides much more understanding of the goals and objectives and it helps them to respond much more ‘personalized’. It also allows them to build a long-term partnership instead of selling the solution and moving to the next RFP. Sometimes it can even help avert RFPs and save much time and effort for both parties.

# The challenge of comparing

I understand this creates a new challenge on how to compare multiple responses. It is much more efficient and easier to compare five lists of checkboxes and count the number of ‘compliant’ checks for each partner. But in my experience, these lists would result in five times a more or less equal number of checkboxes.

It will take more time to digest and compare the responses from partners if you use open questions. I haven’t found a good way of ‘templating’ the responses for efficient processing. But I am not sure if that is even possible for non-functional requirements. As we often ask for attachments like architecture diagrams, implementation plan timelines, certifications, and reports of availability over the last two or three years. And many times, we host demo sessions to see the solution in action or organize a proof-of-concept to build a first use case to prove the potential solution. Processing and comparing this feedback are not black and white but require internal discussion and feedback.

Just plan and prepare to digest all the responses and organize internal discussion sessions with the selection team with both business and IT stakeholders. Agree upfront on what aspects the partners and potential solutions will be rated and create a template for that, fill them in individually, and then discuss.

If you have any tips or tricks on processing multiple partner responses and comparing them, please let me know, and I will include them in this article.

# Using a scoring matrix

I was involved in a European tender at the Dutch Railways where we used a scoring matrix to compare the proposal from multiple software partners. Upfront, we discussed the criteria we would rate the proposed solution. We would score on three levels: insufficient, sufficient, and excellent. For each level, we defined the number of points to receive. For insufficient we discussed if it will be knockout or penalty points, like -10 points. For sufficient we used 1 point and for excellent 3 or 5 points. We communicated this scoring matrix in the RFP so it was clear for partners how they would be rated. Also, the result of the scoring was communicated, although anonymized. So, we mentioned the score of the partner plus the score that the winning partner achieved without naming the winning partner.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Partner A** | **Partner B** | **Partner C** |
| Criteria 1 | 1 | -10 | 1 |
| Criteria 2 | 1 | 1 | 5 |
| Criteria 3 | 5 | 1 | 5 |
| Total | **7** | **-8** | **11** |

I understand this might be an official process for European tender, but it has some good points. It helps think and agree on upfront on what are the selection criteria. And it brings clarity and transparency to the selection process for partners.

## Relative scoring

An idea of improvement I have is to use relative scoring combined with the above scoring matrix. When for example you have 5 partners responding to the RFP, you divide for each criterion 1, 2, 3, 4, and 5 points. This reduces the chance of having multiple partners reach the same score.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Partner A** | **Partner B** | **Partner C** |
| Criteria 1 | 2 | 1 | 3 |
| Criteria 2 | 1 | 2 | 3 |
| Criteria 3 | 3 | 1 | 2 |
| Total | **6** | **4** | **8** |

I haven’t tested this idea in practice, so please let me know what you think of this idea.

# Examples of open questions

Please find below a list of open questions I have used in RFP’s. Most of them are just rephrasing a non-functional ‘requirement’ and some or coming from architectural principles.

If you have open questions you believe I should add to the list, please let me know via the contact form below.

## (Data)Privacy

* Please list the data objects required for the solution to properly operate.
* Please list the data objects required used for your organization.
* Please list the data objects that are shared with 3rd-party suppliers.
* Please describe how data is processed according to EU privacy regulations and law.
* Please describe where the data objects are physically stored.
* Please describe the measurements available to protect customer data.
* Please describe how customer data is protected.
* Please describe how the data model of the solution is open, flexible, and follows standards.
* Please describe how access and use of data are logged, monitored and traceable.

## Security

* Please list the certifications available related to IT security.
* Please describe how the solution is tested for IT security.

## Architecture

* Please describe how the solution is omni-channel.
* Please describe the ideal position of the solution in our IT landscape.  
  Please enclose an architectural diagram.
* Please describe how the solution provides insights into customer interactions with the solution.
* Please describe the (cloud) infrastructure and hosting model of the solution.
* Please describe how the solution can scale horizontally and vertically.
* Please describe how the infrastructure costs are monitored and made available.
* Please describe how you can help us optimize infrastructure costs.
* Please describe what environments are available for development, testing, and production.
* Please describe the deployment model of the solution for feature updates and security patches.
* Please describe the Strength (Unique Selling Points) of the solution compared to comparable solutions.
* Please describe the Weaknesses of the solution compared to comparable solutions.
* Please describe the required business and IT skills to work and maintain the solution. And how you can help us train and enable.

## Integration

* Please describe how integrations with the solution can be monitored and tested.
* Please describe how the solution can integrate with our existing CIAM solution for customer identification.
* Please describe how the solution can integrate with our Active Directory to support single sign-on identification for employees.
* Please describe how the integration can support (near) real-time data transfer.
* Please describe how the integration can support event-based data transfer.
* Please describe how the integration can support batch data transfer.
* Please describe how the APIs available are following standards.
* Please describe how documentation of the APIs available for our developers.

## Service Levels

* Please describe the process for filing incidents and feature requests.
* Please describe the model for governance and escalations.
* Please describe the available service levels and support windows.
* Please describe how you ensure quick response and resolve times (for high-priority incidents).
* Please describe the communication model for high-priority incidents.
* Please describe the service level reports available.

## Implementation

* Please present (in PowerPoint or similar) an Implementation Plan including
  + A project plan, including phases (MVP, Phase 1, …), milestones, and critical path
  + A communication plan and structure
  + A list of potential risks and dependencies
  + An overview of the total team, including ANWB members, and their level of expertise and capacity
  + A plan for data migration
  + A plan for hyper-care (the first 4 weeks after go-live)
  + A list of prerequisites for our organization before the project can start