

DEFINING BUSINESS REQUIREMENTS: This is the process of describing what you want your software system to do, in business terms. We have found that our clients know what they want (their system to do), but are often challenged to describe their needs in software development terms. This 10 Step Guide is meant to help non-technical (and even technical) people define their initial requirements, which will be used to construct a QuickBase prototype, the first step toward deploying your QuickBase Solution.

1. **BUSINESS STORY:** What problem(s) are you looking to solve, what are your business issue(s) and how did you get here? A plain English explanation to these questions provides (your consulting team with) context for defining a technology solution that fits your business. Additionally, we like to use the business story during training (and even embed aspects of it, in your QuickBase application) to explain to users how to use the solution and why you built it, in the first place.
2. **USERS:** How many people will be Users of your QuickBase system?
 - a. Will all Users have the same rights (view, add, edit, provision new users, create new views/reports, admin rights and developer credentials) or do you foresee needing to support different levels of permission (roles) in your application?
 - b. Will Users be from different organizations? Do you need to Hide Views or data from certain User roles?
3. **DATA:** Provide examples of how you currently manage your business data and processes. Explain existing tools and processes (paper, print-outs, MS-Excel worksheets), the quantity of data being managed and the types of data values being managed (data types, restricted value fields – field values that could be limited to a pull-down list).
4. **DATA RELATIONSHIPS:** Whenever possible describe data relationships in parent-child terms. Examples include: a contact management system that records contact information (in one table), and supports relationships that link each contact record to one or more activity records; or where you have an order management system that supports a relationship where an order may have one or more order detail records. This aspect of requirements defining is referred to as the relational aspect of database design. Many people make careers out of database design; so don't get disappointed if these requirements are hard to articulate.
5. **DATA ENTRY:** Will users be required to manually enter data (using web-based data entry forms), cut and paste data from spreadsheets or will there be no need for data entry by users (data bulk loaded by Administrators)?
6. **DATA ENTRY FORMS:** Will all users have permission to enter data into all fields? Do users tend to work on one row at a time or are there scenarios where you need to enter data into multiple rows/records at once (like a spreadsheet)? Is there a need for conditional data entry -- are there cases, based on a data field being of a certain value (Client Type = Prospect), that subsequent data fields/sections may or may not be required as part of the data entry form?
7. **VIEWING DATA:** A key strength of QuickBase is that it provides many different ways to view data (grouped, summarized, user query-able, cross-tabs, charts and even Gantt charts for Project Management). Determining how you want to View (or Report on) your data is one of the more challenging aspects of building a QuickBase system. The best way to define your Views (and the underlying queries) is to write out use cases (business scenarios for Viewing the data), in plain English, which detail what QuickBase data you wish to View. **Use Case Example:** At Monday morning sales meetings, we need a View that allows us to select a particular sales person, the Opportunities they're working on (that are not closed or cancelled), that have an expected close date before 9/30 -- displaying the following fields: District, Prospect Name, Prospect Contact, Prospect E-Mail, First Contact Date, Expected Close Date and Deal Amount.
8. **E-MAILING OF DATA:** Will there be a need to "push" or e-mail users data from the system? Examples include:
 - a. Notify users (via e-mail) when records are added, edited or modified in QuickBase
 - b. The ability to "subscribe" (via e-mail) to view data on a scheduled basis
 - c. Alerts and reminders (via e-mail) that are triggered from date fields in QuickBase, (i.e., send a reminder to field Assigned To, when records with a Due Date that are within one week of today's date)
9. **YOUR QUICKBASE TEAM:** Who are the "players" in your organization? Who should be involved in detailing requirements? Who will be involved with the development, testing and the acceptance process (the people who say, "This is good to deploy as version 1.0")?
10. **DEPLOYMENT PLAN:** How will you train your team? Will you require documentation of your QuickBase solution (to provide to users and keep internally in your organization)? Who in your organization will support users when they have questions and serve as the conduit for getting answers from Intuit or your consultant(s)? Who will track bugs and feature requests for subsequent versions, and document your next set of business requirements?