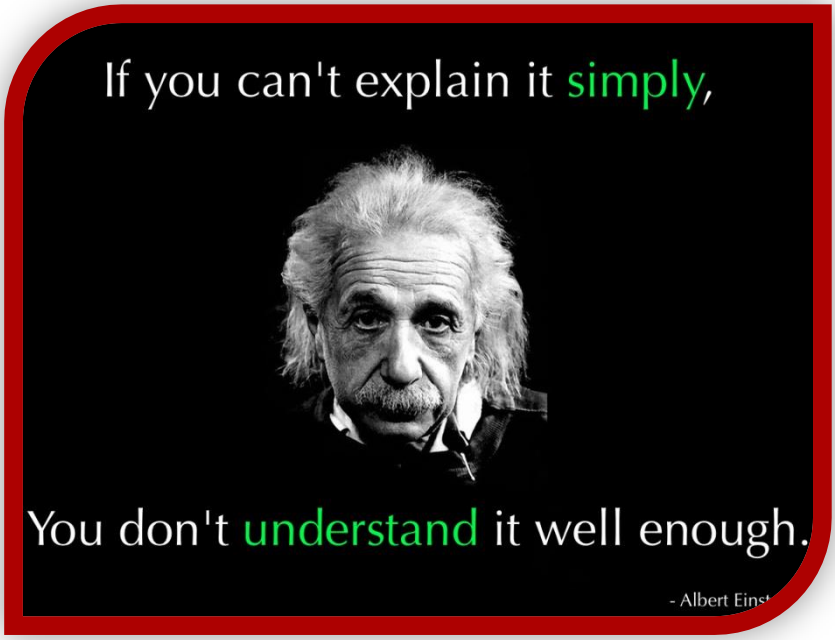

DRAGONPOINT
SOFTWARE
Custom Software or Off the Shelf Package?
10 Questions to Identify the Right Solution!

Executive Summary

A software requirements specification is written to give the company that's paying for the project, the people who will use the system, and the developers who will be writing code a clear shared picture of what an application is supposed to do.



Because this can be a time-consuming task, companies are tempted to think they're saving money by skipping requirements. Don't make this mistake! Defining what you need before writing code is the best way to meet your business objectives and keep a software project in budget and on time.

Photo credit cover page *@Doug88888*

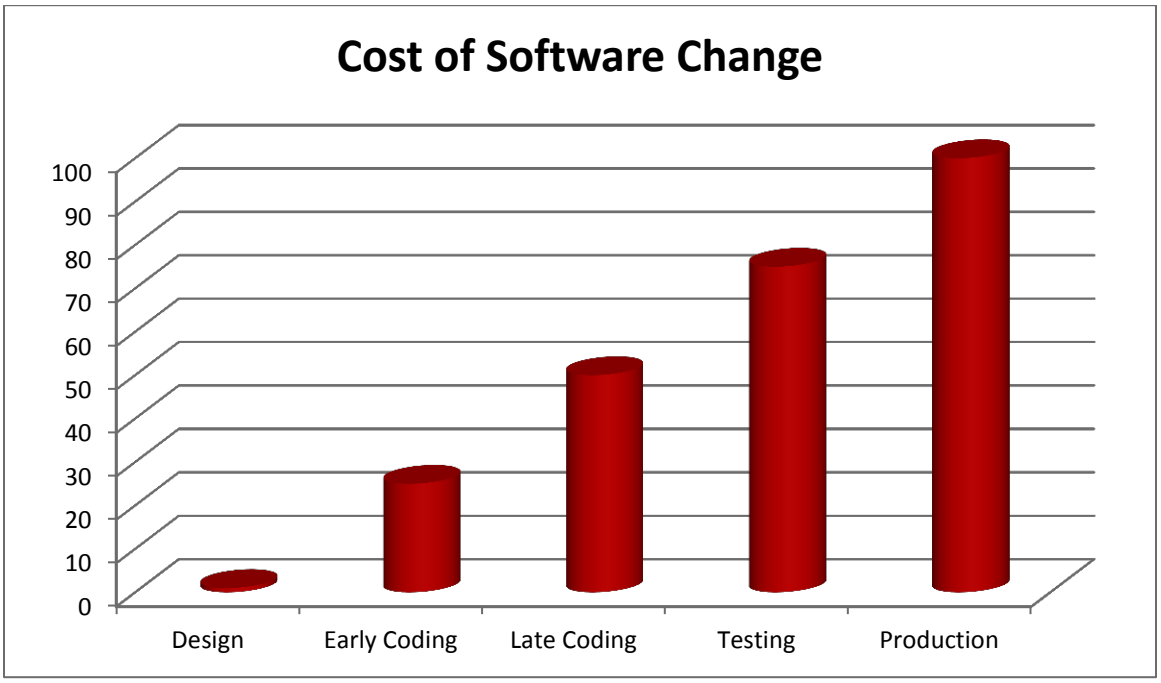
Photo credit this page *ransomtech*

Table of Contents

Is a Requirements Document Necessary?

It takes time to develop a software requirements document. The subject matter experts and the requirements analysts have to identify the business needs, define the software that will meet those needs, and put everything in writing – and this is almost always an iterative process.

Some companies try to save time and money by going directly from business need to coding, but this is a costly gamble. Why? The later in the project you identify a change, the more expensive it becomes, as illustrated in the chart below, which shows the relative cost of a design change at each major phase of the software development process.



Whether you're partnering with an outside firm or relying on your internal team, the following steps are critical to creating an understandable and usable software requirements specification.

If the software you need includes more than one module or you're overwhelmed at the idea of defining all the requirements up front, break the project into smaller pieces. You can define the requirements for phase 1, and while it's being coded, you can start on requirements for phase 2.

1. Understand the business process

Before you can write requirements, you must understand the specific way the business works. Although businesses in the same field may have many similarities, one size does not fit all. Each business has unique characteristics and processes that ensure its competitive advantage. Include a description of the business process in the specification. Words are good, and pictures are better.

2. Speak the language

A good specification is written in the company's business language. If a client speaks German, it wouldn't be productive to send her an email written in Greek. In the same way, if Bob's Manufacturing, says, "We maintain the lowest possible level of parts in our inventory," his specification should use the word "parts" not "items," and the system should have a Parts, not Items, page. Although certain words may seem to be synonyms, company-specific terms are not interchangeable because they're part of the language and culture of the business.

3. Draw pictures.

Many ambiguities are eliminated with pictures. Companies understand information when it's in a format that makes business sense. One of the best ways to ensure the company and the software developers have a shared understanding of requirements is to include screen mockups. To make the drawings even more useful, the requirements document should explain how each screen will work, including the way information gets into each field and what happens when you press every button.

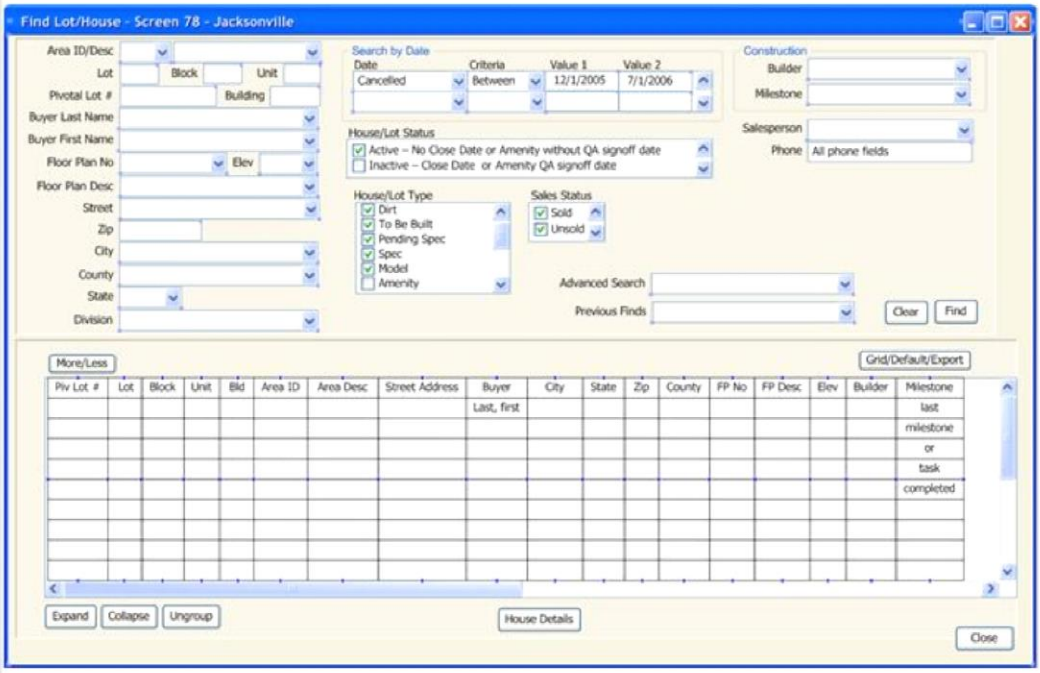
Find Lot/House– Screen 78

The Find Lot/House screen (78) is used to find and display information about lots and/or floor plans being built on lots (regular sales or specs).

Access the Find Lot/House screen (78) by choosing the Purchasing main menu option, House Sales Budget sub-menu option.

Find Lot/House is also available as a Favorites selection.

Select the criteria at the top of the Find House screen (78) and press the Find key to display matching records in the results set.



To ensure results include only one row per house, the buyer, salesperson, and phone columns display only one record. To view all buyers, salespeople, and phone numbers, click a row in the grid and press the House Details button.

Entering a zip code displays the associated city and state. If the zip code is unknown, choose a state and then a city or zip code. Choose a zip code and enter an asterisk (*) in the city field to see all cities for the selected zip code.

You're an experienced specification writer?

Don't be tempted to impress subject matter experts by telling them how much you know about what they do. If you're asked to explain how other companies have solved similar problems, don't let your answers discourage new ideas. Even if you've written 100's of specifications, do not assume you know the unique business process better than the people who do the work every day.

4. Talk to the people responsible for the work.

The only way to figure out what a system needs to do is to talk to the people who are responsible for the work. The discussion should include lots of questions, and the requirements analyst should do more listening than talking. This leads to the next step . . .

5. Listen to the people responsible for the business process.

This may be the most important thing the person extracting the requirements can do, but it's also the one most often ignored: LISTEN.

An added advantage of listening and incorporating what you hear into the new system's design: new software means change, and people are more likely to accept - and maybe even embrace - change when they help create it.

6. Invest enough time.

A good specification includes a lot of information, and it takes a significant investment in time to create it. The work to be completed includes:

- Discussion between subject matter experts and requirements analysts
- Draft requirements including screen mockups
- Review draft with subject matter experts
- Incorporate changes and finalize document
- Review and approve document

The time investment depends on many factors including the complexity of the system and the experience of the subject matter experts and requirements analysts, but it could require four to eight hours to create a complete specification for a small application with six pages/screens. On the following page, you'll find guidelines to help you estimate the time it may require to develop specifications for a larger project.

How long does it take?

Assuming the work is completed by an experienced analyst who has written many specifications, here's an estimate of the time required:

- Initial meeting: 1-2 hours per screen or function for discussion between the subject matter expert and the requirements analyst.
- Draft requirements: 1-2 hours per screen or function for the requirements analyst to transform the rough information into page mockups and organized requirements.
- Review draft: Approximately 1 hour per 4-6 screens for the subject matter expert to complete a detailed review of the requirements and identify questions and modifications.
- Clean up document: 1-2 hours per screen or function for a follow up conversation between the subject matter expert and the requirements analyst to clarify understanding and discuss possible modifications and alternatives.
- Finalize document: 1 hour or less (depending on the number of changes) per screen or function for the requirements analyst to update and finalize the requirements document.
- Approve document: 1 hour per 3-4 system screens/functions for the subject matter expert and management to approve the requirements document.

This is a timeline based on an ideal world where there's one subject matter expert who knows how the process works and how the new system should work. Any of the following conditions extends the timeline:

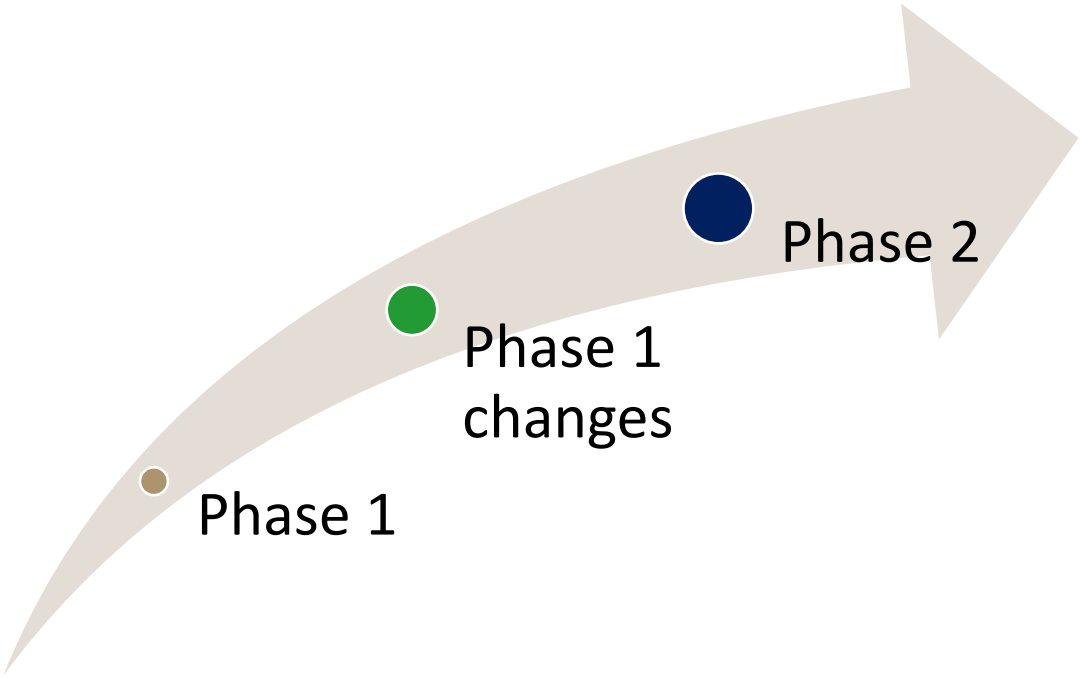
- More than one subject matter expert.
- The existing process is being replaced, the new process is being created during the requirements definition phase, and no one has a clear, detailed picture of how the new process will work. (This happens more often than you might think!)
- Any step is iterative. For example, add additional time if the subject matter expert and analyst need two or more meetings to discuss a screen or process.

7. Don't assume a specification is complete.

No matter how well you define requirements, how clear the process is, and how accurate the screen mockups look, when people actually begin to use it, they will come up with new ideas or find that even though the system works as designed, it's not exactly what's required. Don't be surprised when users find holes in processes that everyone thought were complete and discover problems with assumptions that seemed sound on paper.

Changes mean modifications to the specification. A specification is only "final" when the new system is working in production.

Note: This is a very good reason to break a big system into smaller, manageable pieces. If you include one typical screen in Phase 1, you can see how a mockup translates into a working page, and you can incorporate modifications to Phase 1 into the next phases. This iterative approach is a great way to minimize surprises at the end of a project.



Save time and money with a good specification

A coherent specification saves time and money because it eliminates costly rework, and it improves satisfaction because you know what to expect.

Following these steps doesn't guarantee you'll get a requirements document that actually communicates clearly – that depends in large part on the skills of the people involved. But ignoring these steps increases the probability that your new software will take longer than expected, cost more than planned, and fail to meet your expectations and business needs.

What does a BAD specification look like?

A client says, “I make money by selling coffee to employees. So, every day when an employee boots his computer, display a picture of a big blue cup of coffee. If it’s before 8 a.m., also show them a digital clock with the current time, so they know they have a few minutes to buy a cup of coffee before they start work.” Unfortunately, the specification often ends up being undecipherable as in the following example.

Nonfunctional requirements: Hex triplet 0000FF recommended for user class “coffee drinker” layer, liquid beverage object. Undetermined hex triplet indicated for numeric output subsystem, though something in the range of 808000 may be architecturally sound. Note: Using the hex triplet allows for more than 16.8 million colors. [The hex triplet is a six-digit, three-byte hexadecimal number.] To obtain the hex triplet value, convert the decimal RGB value, usually 0-255, by dividing number by 16, ignoring remainder, to get the first hexadecimal digit 0-F, where A-F represents 10-15 (however, if original number is 0 or 1, multiply by 255 before conversion); second digit is the remainder times 16; repeat process for each of the three RGB values.

Functional requirements: Add program to initial boot sequence to ensure application is automatically instantiated on load. Numeric digital output to be clearly readable based on standard 20-20 scale and should derive value from date/time default settings on local system; specific pixel size for “box” (square? rectangle? irregular shape?) to be determined, as is location of communications interface to be displayed in foreground of “box.” Size of container for desired end product to be determined, though recommended standards are specifically 8 fl ozs, 12 fl ozs, and 16 fl ozs. Note: don’t represent non-standard size for customer’s product, because if representation should drive demand, the non-standard size may be obtained only with additional unspecified lead time and at a premium cost.

About DragonPoint

DragonPoint has 25 years of experience developing software requirements documents and software systems for businesses.

Call DragonPoint today at 877-542-0657 to talk about your software needs and get expert assistance writing a specification that communicates your business needs!