SonicSpider LLC.
www.sonicspider.com
P.O. Box 483, Bonsall CA, 92003
Ph# 619.955.6380
Fax# 760.453.2177

Custom Web Development Guidelines and Policies

Small Projects

Table of Contents

Introduction

Custom web development involves a partnership between the architect/designer/programmer/developer (SonicSpider) and the owner/testers/users (your company, the CLIENT and your customers) of the site or application. The website or web application evolves from a statement of need by you the CLIENT, to a general specification, a prototype, growing "work in progress", and ultimately, a finished website or web application. You will invest considerable time and money and want to get both what you want and what you need.

Most likely you have not been involved in very many custom software projects and you may be feeling a bit lost. Or, if you have gone down this road before, it is possible that it did not turn out to your liking. Why is that?

Generally it is because the person or company you hired may have been proficient in the technology, but was clueless on how to manage a successful project. This may seem harsh, but sadly most people that sell themselves as "web designers or web developers" don't know how to run a project. They may have what are call splinter skills in design, marketing, or have experience coding but not have the depth of knowledge and experience to see the larger picture. Having isolated skills are nice, but having only a hammer does not build a house.

SonicSpider is in the unique position of having over 20 years of experience in project management as well as a team of experts that cover all of the necessary skills to build the components of your web project. Many of the principles of SonicSpider have worked with diverse teams of experts over the years and therefore have a depth of experience that is hard to match.

This is a special version of our guidelines for small projects. The purpose of the guideline is to inform all members of the development team, which includes you, of what issues to expect to encounter and what is expected from of every member of the project. It also provides the necessary structure and protocols for success. It will act as a guide on the core methodology for a successful project and is considered part of your proposal or work description.

The ultimate goal it that you, the CLIENT, get a website or web application that exactly meets your needs, on budget and on time.

Expectations

Most of us browse the web often enough that we develop an expectation on all of the different things that can be done on a website. Many of these websites are built by large corporations that have web development budgets in the tens or hundreds of thousands of dollars, with a staff of in-house web developers working full time. As you use and browse these websites, it is very hard to get a feel for the costs, both in time and money, that are involved in building and maintaining these sites. These costs are not advertised and certainly very hard to find out after much research. Based on a number of industry articles, it is not uncommon for corporations to average about $10,000 per page. Yes, that is **ten thousand dollars per page**. Obviously with that kind of investment you can do a lot of very cool things. Given that number, what should our expectation be for, let's say, $100 per page?

Fortunately for you, a fair amount can be done, but there will be a lot of compromises and hard choices. Some pages will cost more, some a bit less. Basically it comes down to focusing on your priorities and then deciding on the compromises that will be needed to get closer to what you want and stay within some rational budget. What you want is to feel that you are getting the most out of your budget, but you have to make some hard choices.

The goal of your design and development team is to help you articulate you needs and requirements and them find the closest match between your budget and meeting as many of those requirements as is possible.

Despite the best of efforts of your team, there still can be some disappointments. Some of what you expect is often not clearly articulated or evolves quietly from the beginning of the project to the very end. As you see parts of your website come to life, you start thinking of these other ideas and needs. Often these “silent expectations” , especially what seems to be small ones, get added to the project unconsciously without the rest of the team knowing anything about these changes. This spells trouble!

When this happens, it can become a very frustrating situation for everyone on the team. It is frustrating for you in that you may feel strongly that this feature or component is a “must have or obvious” and should be included. On the other hand, your developer/designer team members find themselves spending extra hours that were not part of the proposal. For better or for worse, this is a business, not a charity, so you start getting reminded repeatedly, “that will cost more”. The project begins to spiral out of control and your patience and the budget become the first casualties.

Sadly, bad feelings and possibly angry words can come from the expectations not matching the reality of your project. The best way to avoid this is to accept that it can and will happen, and manage it right up front.

Expectation Management

The foundation of managing expectations centers around how well the project specification (in the form of a proposal or work description) articulates exactly what you envision. **First and foremost, you need to understand that his is nearly impossible to accomplish.** This is because we all take the words we read and fill in our own “blanks” that fit what we want or think we want. Also, it is human nature to “skip past the obvious” and assume that everyone is making those same assumptions. This simply does not always happen and is best to never expect it to happen.

But despite that rather dismal prognosis, there are a number of guidelines we can follow to maximize the possibility that all members of the team have the same “expectation”:

1. If it is not explicitly on the proposal or work description it will not be included in the project.
2. If you expect something it must be included and clearly articulated.
3. Nothing is assumed, standard or trivially obvious.
4. Ask questions and clarify everything.
5. Just because it may be commonly available on some or many other websites, does not mean you will get the same. (See #1)
6. Changes or additions that are not on the proposal or work description are extra billable items, therefore it is always best to get them included in the initial proposal or work description.
7. Patience – Remember we all have the same goal!

That goal is to maximize what you get and stay on budget. To do this your expectations must be possible within that budget, and be clear to all other team members. Keep this in mind as the project unfolds and we can help you get your expectations met.

Definitions and Terms

There are a number of terms in the web development business that can be unfamiliar or used incorrectly or interchangeably, often to the confusion of you the CLIENT. Here are some commonly used terms and a brief discussion of each that will help insure that everyone understands the remaining discussion. Please review these terms and refer back to them from time to time.

- Web Page – A single page with a unique address (URL – Uniform Resource Locator). There is no limit as to the length of a web page, but given that users generally dislike to scroll, it is a good idea to keep the page as focused and close to a single “screen” as possible. Operationally, we define a single web page as being LESS then two printed pages.
- Static Website – A website based on a fixed set of web pages that do not change without direct editing of a file, by you or someone you hire. These pages are linked in a logical fashion. There is generally a menu or navigation mechanism.
- Dynamic Website – A website that is generated from the contents of a database or other information storage format (XML, configuration files, templates, etc). The pages are generated on demand by a script or other code execution process. Dynamic sites will typically alter their appearance based on user input, this makes them more expensive than static sites.

- CMS – Content Management System – This is a dynamic site that is built such that you the CLIENT can change or edit the webpages and the structure of the website. This can be very tempting to the small business, but it does involve a lot of time and learning to do this effectively. You need to seriously decide if you have the time to build and manage your website and also run a business.
- Web Design – The general process of designing the look and feel for a website. This generally exists as a "picture" and is done using various applications like Adobe Photoshop or Illustrator. Samples are generally sent to you as either a "jpg" or "pdf" file, but the actual file format is either "psd" or "ai" and require the actual program to view.
- Needs Assessment – The process of determining the exact needs of a company with respect to their website or web application. This is a commonly neglected step in the process and that neglect can directly be attributed to the failure of a website to meet that needs of a business.
- The Design - The end result of doing a "web design". This is the final "picture" that you approve and is used as the specification for determining if the final website meets that specification.
- Design Breakout – This is the process of taking the final design and breaking the images, colors swatches, and structure components into the necessary pieces to be used by the HTML/CSS coder.
- The Coder – This is the person that renders your design into actual HTML code. Care must be taken when selecting this person. Poor page architecture and coding can leave you with a mess that is nearly impossible to update and maintain. A good way to insure your website has been coded properly is to run your webpage through the "validator" provided by the "World Wide Web Consortium" or "W3C" for short. (http://www.w3c.org)
- Page Layout – The relationships between different components of a web page. Many designs have similar "layouts". Common components are: header, footer, sidebars, main content, menu bars or navigation blocks. Often the "home page" has a slightly different layout than the "interior pages". This is because they often have different purposes.
- Content – The main text or textual purpose of a web page.
- SEO – Search Engine Optimization – Making the web page easier for a search engine to index and determine when and where to include it in search results.
- Web Development - Often used incorrectly and interchangeably with "Web Design". Generally meant to indicate the full process of which the design is only one component. This the process of project management would include: planning, design, market research, coding, testing, and deployment.
- HTML – A standard markup language that consists of tags that tell the browser how to render the structure of the page. This standard is specified

by World Wide Web Consortium, often refereed to as W3C. The W3C is an international body that has members from all facets of the web industry.

- CSS – Cascading Style Sheets. This is a separate document from the web page that tells the browser how to "style" the structure of that page. CSS is also governed by standards set out by the W3C.
- Coding – Rendering the design to HTML and CSS. The HTML is the core code and includes the structure and content of the page. CSS is the application of the design to that structure and content.
- Web Standards – Building websites to industry standards as specified by the W3C, so that all browsers render the pages identically, and to optimize search engine access and processing. This also includes: usability, accessibility, SEO issues, and maintainability.
- Accessibility – A website that is structured for maximum use by differing means. This includes, sight impaired, color blindness, the blind, and smaller formats like cell phones and PDAs.
- Usability – The use of common page elements and features that have become familiar to the majority of users.
- Cross Browser Testing – Insuring that different browsers display the page in a similar and useful manner. It will rarely look exactly the same.
- The Fold – That point in which your webpage scrolls down below where it is visible when the page first loads. A key design concept is to make sure that your visitor's attention is captured quickly by that part of the page that is "above the fold". Your visitor will not scroll the page unless you give them a good reason. How much of your page is "above the fold" will also depend on the "resolution" of that user's screen display.
- Resolution – A measure of how the computer screen displays the details, colors and size of all windows, text and images. There are many resolutions depending on the equipment of the client of customer of that website. Currently, about 80% of web users have a resolution of 1024x786 pixels or greater. The majority of users (36%) use 1024x786 pixels.
- Third Party – Software, plugins or code provided and supported by others.
- Hosting Service – This is the provider of your hosting services. Examples are GoDaddy, Enom.com, EarthLink, Network Solutions and others. Each service has unique procedures and policies for managing their services and these procedures and policies are subject to change without notice.

These are a lot of terms and concepts to remember. Make a point of reviewing this list several times and ask questions. The better you understand these terms and concepts the better "team member" you are of the web development process.

Third Party Software (TPS)

It is not unusual for your project to be composed of some third party software. This can be in the form of complete applications, modules, plugins, extensions, libraries, or scripts, these will be referred to as “applications”. In all cases TPS has unique issues, constraints and limitations that you should be aware.

Installation

The installation of the TPS will be outlined in your proposal or work description. Generally this involves the dependency of various configurations on the Hosting Service (See definitions). Each Hosting Service has unique installations of various services to which your application may require. Because of this there are the following limitations and constraints:

1. SonicSpider is not responsible for configurations and services provided by the Hosting Service.
2. Documented requirements for your TPS application may not cover all requirements as the authors of the software may have made assumptions that are not documented. These requirements may not be provided by your Hosting Service.
3. Your Hosting Service may not provide the proper services or environment for your application. In which case all extra time or work required to work around this issue becomes an extra billable expense.
4. In some cases your application can not be installed. All time spend discovering this condition is billable as part of your proposal or work description whether it is stated explicitly or not.

Configuration and Setup

Your proposal or work description will outline the details of the configuration and setup requirements. It is important that you understand that most software has considerably more settings than will be addressed in your installation. SonicSpider will be limited to those settings and configurations that are explicitly outlined in your proposal or work description. Some settings are made in configuration files and therefore are not generally accessible to you, whereas others are part of an administration panel or console which you will have access to and the ability to alter at a future time. The following limitations and conditions apply in all cases:

1. SonicSpider is only responsible for those configurations and settings explicitly outline in the proposal or work description.
2. Settings or configurations that are accessible to the CLIENT are no longer the responsibility of SonicSpider after such time as the CLIENT has accessed that part of the application (Administration panel or console)

3. Changes to the behavior of the application due to upgrading or the addition of extensions or plugins may require additional billable time to correct or adjust.
4. Once you, the CLIENT, have accessed the configuration files or administration panel or console, SonicSpider is no longer responsible for those settings.

Upgrades, Plugins and Extensions

During the life of your TPS, there may be upgrades or enhancements to that software. Generally upgrades are advisable as they often address security vulnerabilities which would allow for malicious attacks to your site. Unfortunately this upgrade or enhancement process can alter the software behavior and of make it cease to operate. This means that you should always backup your installation before making any changes and also adds to the time or cost of doing those upgrades or enhancement. You should be aware of the following:

1. Upgrades or enhancements can alter or render other add-ons or plugins non operational.
2. Upgrades or enhancements can alter or render configurations and styling settings non operational.
3. The time and cost of correcting these issues is always an additional expense beyond that which is listed in your proposal or work description

Development Process Overview

Custom development involves four basic phases:

- Needs Assessment
- Design
- Programing
- Quality Assurance (QA) (and Documentation if specified for a dynamic site)

Of these phases only the programing phase can be quantified with a cost estimate to any exacting degree, and ONLY if the needs assessment and design phases were through enough to allow for a detail specification to be developed.

It is tempting to minimize or skip the needs assessment and design phases, especially in small projects. The typical attitude is that “we will quantify our needs as we go” or “..it is so simple that it should be obvious” or variations on those feelings. Interestingly, at the same time an “exact” or “good approximation” of the cost is expected. This leads to a contradiction: A good estimate of an unknown. It is very simple; ONLY if there is a detail specification can there be anything approaching a “good approximation” of the cost. Detail specifications are developed during the needs assessment and design phases. The design phase typically follows the “needs assessment” and takes these “needs” and turns them into a specific design and structure for you website. These two phases are often lumped together as a single process. This is fine, but remember they are two distinct steps and make sure you address both.

A Rule of Thumb

The “rule of thumb” on both the time and cost of a software project is: *“For every hour of programming, there will generally be about, one hour of design and two to three hours of QA and documentation”*. Ironically the programming phase depends on the design, and with out the design it is impossible to generate an estimate. This is because it is dependent on needs and requirements that are only discovered as the planning is being developed. It sounds like a vicious circle, which it can be at times, hence it is often referred to as the “bootstrap” phase of the process.

Step I - Needs Assessment, Design and Planning

Would a house be constructed without a detailed blue print? Maybe a very small tool shed, but not a house. You can't run a business in a tool shed. Architects expect to get paid for their designs and engineering. Software is no different. On occasion, the desired project centers around an existing website or application that is to be re-written and/or expanded. On other occasions, the needed website is completely new. Either way the documentation of the expectations of the targeted users and the business needs of the company must be clearly presented. The key concept is that “**If it is not written down, it will not happen**”. There are NO assumptions of “commonly accepted functionality”. The obvious must be included and documented clearly. Some projects are expected to “evolve” and features are added and documented in increments. Other projects require a clear “feature complete” specification. Either way, each feature, each change, and each enhancement must be clearly documented. Typically, one quarter to one third of the project time and cost, will be spend in some form of “Needs Assessment, Design and Planning”

It is important to repeat and emphasize: “What is not clearly specified can not be cost estimated”. Even if the project is meant to “evolve” the design and planning step must be completed before the next increment of the project can be estimated.

Needs Assessment

It is very common for small projects that you feel confident that you know exactly what you want. Unfortunately, this has rarely turned out to be the case. Some common things that you might not think of for your website or web application:

- Static Websites
  - SEO, accessibility, and usability issues
  - Internet marketing issues
  - Image quality and download speeds on typical bandwidths.
  - Browser "fold" and user monitor resolutions.
  - Browser versions compatibility
  - Client hardware – Monitors, screen size and resolution.
- Dynamic Websites
  - Error conditions
  - Defensive code for anti-hacking
  - JavaScript degradation (Some users turn that off)
  - Plugin requirements (The ubiquity of the plugin or lack of).
  - Hosting and database requirements
- eCommerce Websites (Extension of Dynamic Website issues)
  - SSL Certificate
  - PCI compliance (Security)

This is a very short list and there are dozens of other issues that can come into play depending on the scope and purpose of your website or web application. It is not unusual that your would not be aware of these issues but none the less, they are very important for a successful website. The purpose of a needs assessment is to understand what your core needs are and then educate you on all of the "hidden" issues that come with executing your core needs. From that assessment, a plan and design are developed.

Use Cases – Dynamic Sites

A very specific part of the design process deserves special mention for dynamic sites. Most dynamic web applications are intended for people to interact and accomplish specific tasks. What "happens next" depends on the input of the user. Because of this fact, the development of "use cases" is the key to insuring that the applications will be "useful".

Use Cases are short "stories" that outline in a "dialog" fashion how a typical user will accomplish a task. These "use cases" allow for the validation of page sequence, field sequence and instructions found on any given page. During the QA phase these same "use cases" will be used to guide the testing and validate that the final application does in fact: "work-as-designed".

This is a critical step and will seem obvious and trivial. Unfortunately it has been SonicSpider's experience that the way a typical user will use an application is rarely that obvious or trivial.

Design and Architecture

The design and structure of your website is determined in this phase. There are two basic components: The actual design or "picture" of your website and the structure of your website. The "picture" part is easy to understand because websites are very visual, but the structured can be harder to understand because there are many possibilities and often there is no "right" structure. Both are important and must support a number of core principles that an effective design should address:

1. Define your target audience – your design must appeal to them, NOT YOU!
   1. Integrate your marketing strategy into the design.
   2. Research what your target audience is familiar with seeing.
   3. Understand your target audience's needs, fears and desires.
2. Be clear on what you expect your website to accomplish.
   1. It must be part of your business plan.
   2. It must receive the attention it deserves.
   3. It must support your goals and objectives.
3. Don't make your audience think!
   1. Users are looking for quality content.
   2. Users scan first, then they might read.
   3. Users are impatient, remember the 6 second rule: "If they can't see what they want in 6 seconds, they are gone!"
   4. Understand what "pushes their unconscious buttons".
   5. Don't beat around the bush.
   6. Make it clear where they are to go next in your website.
4. Don't try their patience, get to the point FAST!
5. Help them focus their attention. Keep it simple!
6. Have quality writing in your content.
7. White space is your friend. Don't be afraid of it.
8. Use a visual instead of words whenever possible.

Step II – Programming and Coding

Once the planning and specification has been completed for either that phase of the project or the entire project, the programming portion can be more accurately estimated. The degree of accuracy is directly proportional to the degree of specificity of the design. Often in small projects it would seem that the programing would be quick and simple. We have all seen movies and TV programs where the computer wizard bangs on the keyboard for a few seconds and then saves the universe from all manner of evil. The reality is so very different. The bottom-line is that “computers are dumb”, and because they are dumb, they need to be told every possible issue and every possible alternative in advance. They can not “think on their feet”. An example might be helpful: Let's say that we want our web application to add the price of several products and calculate the tax. Simple right? For you and I, yes, for a dumb computer, not so simple. What happens if:

- A product price is missing
- The tax is not defined
- The database or configuration file that defines the product or tax is not available.
- etc...

As you can see there needs to be programing for each an every contingency. A “simple” one line calculation now becomes twenty or thirty lines of code with additional code for handling each and every possibility. Then each has to be tested to make sure they handle the problem correctly. Ultimately that single calculation can require dozens of lines of code and hours of testing. In the mean time, the universe was destroyed.

Generally, even the most simple of tasks can require several hours of programing and testing. By programing defensively we insure less bugs and less problems later, but that takes time, there is no way around that fact.

Step III - Quality Assurance (Testing, Debugging and Approvals)

Depending on the type of site being developed for you there are different aspects of that development that need to be either tested or “approved” before the project is considered completed. This is also where the original specification comes back into play. That specification is used to insure that the site looks and behaves exactly as your requirements specify. If the specifications are vague or incomplete then you can see that this stage of the process can have problems. At what point is it “done”? How would you know that you are getting the end result for which you have paid? The specification that seemed so obvious and trivial to you early in the process now can become your lifesaver because it protects you by insuring that you get what was agreed.

This general process is called “Quality Assurance” (QA) and it is your job as the CLIENT to **assure** that your website, be it a static, dynamic or eCommerce site meets your needs. QA is not something that happens JUST at the very end, it is a constant process.

SonicSpider will provide you with many opportunities to review the progress of your site development. At each of these opportunities, it is very important that you review that progress and quickly report any issues you find. If you have questions or don't understand what to expect, ask. Generally you will be instructed as to what part of the design specification you are to review. You should keep your focus on that part of the specification and ignore other parts that are not yet fully functional.

QA is an allowance expense NOT a fixed expense.

Unlike the programming part of the estimate, QA is not a fixed expense but an approximation, or an "allowance" of the "average time it takes to test and approve and average task". You are part of that process in that you have the responsibility of the ultimate approval. How quickly you provide that feedback and approval can have a big impact on keeping costs under control. Following is a discussion of a number of areas where you impact this process and the cost.

Why is timeliness in reporting problems so important?

Your web development company has many clients and many projects going at one time. Time is scheduled to work on your project, and one of the major killers of that schedule is delays in waiting for a CLIENT's feedback. If thereare delays in feedback, other projects are scheduled in your place, increasing the delays and increasing the likelihood that something will be missed, forgotten or done incorrectly. It is increases the management, increases the possibility of miss communications. The list goes on, but key point is that by not responding in a timely fashion, you are increasing the cost, and making it harder for your web development company to meet YOUR needs.

Why does fixing a bug or problem often cost more money?

The initial programing creates the core logic and structure of your site or application. Every effort is made to code defensively and anticipate problems or issues. Unfortunately, the number of possibilities is dependent the completeness of the specification and for a dynamic site, how someone interacts with your site. Static sites are more predictable and therefore the issues are smaller and less costly. Dynamic sites on the other hand can involve issues that were never documented or specified. This will lead to increased costs. Again this illustrates the importance of the planning and design phase of the project.

What if there are new requirements?

In any size project, it is not unusual to discover that something was missed in the original design specification. Remember: "**If it is not written down, it will not happen**" so the first thing is to review your proposal and specifications. Then clearly outline what it is that you think is missing or needs addressing. The more detailed you are, the less it will cost you to address these additions. Change requirements are extra and if you ask for those changes, you will be billed for that additional work. Also keep in mind that each change will require: A plan or specification for that change, the programming, and then the testing of that change.

What if I find a problem a year later, after my site is finished?

This is not as uncommon as one might think. This generally is the result of incomplete testing and specifications. On static sites this is usually a missed textual error (content, spelling or grammar) or a bad link or navigation issue. All of which should have been caught by you the CLIENT, during the testing or approval phase. On dynamic sites this is much more complicated. It also can be due to poor testing or incomplete specifications but it is not uncommon that it is just because "no one ever thought that anyone would do that". (which actually leads us back to the value of "use cases" discussed earlier). Either way it is an expense.

How can I minimize QA costs and keep them under control?

The answer to this is clearly defined but often ignored:

1. Clearly state your needs.
2. Insist on detailed planning and specifications.
3. Understand the process (these guidelines).
4. Document communications and "close the loop" (i.e. make sure you respond and get a response).
5. Respond promptly and completely as delays will require SonicSpider to spend extra billable time refreshing our understanding of the problem.

Understanding The Development Cycle

Keeping the process under control.

Custom development generally follows a cycle, and you are very much part of that cycle. In order to make the most of this process you need to understand your role. A typical cycle will contain all or most of the following steps:

1. Discuss or review the requirements.
2. Initial programming of requirements.
3. Alpha testing of basic functions (Repeat #2 if needed)
4. **Deliver release to client**
5. **Client reviews and provides feedback**
6. Changes and adjustment corrected
7. Repeat to step #4 until there are no further changes or adjustments.

Steps #4 and #5 are where you come into play. This is also where projects can stall or get off track. It is not unusual that you have many other responsibilities and time is limited. You receive and email from SonicSpider indicating that there is something to review, approve, or test. You are busy, tomorrow or "in a few days" looks better or more likely. Problem is: You just added one or two days to that project. There are many cycles in a project, even the smallest of projects so one or two days can quickly add up to "weeks or months". These delays can have costly consequences.

Timely reporting of corrections or issues.

You, the CLIENT, are a critical part of development cycle and therefore any delays at your end will increase the time it takes to complete your project. There is a danger in those delays: The project and its specifications can become fuzzy or off track. Basically the focus of the project is critical. The programing required for your project requires focus and forcing a week or more delay will require time to "refocus" back on your project. This "refocus" takes time and therefore is a cost. SonicSpider has many projects going on at the same time, and we have project software to help keep track of the status of your project. This helps tremendously on the day to day issues, but delays of weeks or months still require a complete review and refocus. The other problem is that you loose track of what you ask for in your website or web applications. You loose focus as well. This causes you to miss issues or add something that was never specified. This can increase the cost.

Guidelines for Testing and Approvals

Following is a checklist of steps and reminders that should help you understand and insure that your participation as a team member goes smoothly:

1. CLIENT responsibilities.
    1. Print out the email you receive from SonicSpider and place it in a special folder for this project.
    2. Respond to SonicSpider and confirm that you have received the email. Also ask any questions that come to mind regarding what you are confirming, approving or testing. Make sure you are very clear as to your responsibility.
    3. As quickly as possible, make time to review or test the deliverable.
    4. Take notes on the email printout as to problems or changes.
        1. Be exact and specific. Vagueness requires more questions for clarification, this then takes more time.
        2. Note any questions you might have. It may require more information before you can provide better details, ask those questions immediately.
    5. Reply to the first email (audit trail) with your feedback, changes, questions or adjustments. (Note, if there were multiple recipients to the email click “Reply All” so that all recipients are informed.)
    6. Insist on a response to your email. If you do not receive a response, resend. Emails get lost in spam filters, so no response may be because it was not received. Make sure you have closed that loop and that SonicSpider acknowledges receipt of your email.
2. SonicSpider's responsibilities.
    1. Provide details with the deliverable to insure that the CLIENT understands their responsibilities.
    2. Make sure that the CLIENT has received the email.
    3. Review feedback or approvals to insure that all responsibilities have been addressed.
    4. Confirm receipt of email.
    5. Follow up with changes or responses as appropriate.

Training.

Training, if applicable, begins with the first release and is an integral part of QA. A representative sample of targeted users should be using and practicing on each release.

Common Release Definitions

Web application development is handled by specified release milestones and is defined as "any unit of work that is complete enough for a reviewer to test or confirm the usability or adherence of the specification". Releases are provided on a frequent basis so as to provide a tight feedback between the development of the application and the user's needs or requirements. Any given release will frequently be missing features and is NOT intended at any time to be a complete and final application or site, but a "work in progress". This evolution of releases insures that the user's needs are met and avoids surprises in the final release.

Though there are no hard and fast rules as to the types of releases that will be available and in what order, the following is a guideline as to what to expect. In all cases the developed "use cases" and design specifications are used to validate the deployed release version.

Generally there are three categories of releases:

1. **Alpha Release** – This is a release that may not actually "work". It is meant to illustrate "structure" or layout issues. In the case of static web pages, it is common that text is there ONLY as a place holder unless you have provided the final copy. The purpose of an Alpha release is to confirm that we are on the right track. Pay attention to the request for feedback and don't get side tracked on issues that will be provided in the next releases.
2. **Beta Releases** – This are releases in which some portion, module or series of pages are working sufficiently that it can be tested. When a Beta version is released it will also contain some form of instructions as to "what is to be tested or reviewed". There will be ANY number of Beta Releases. Some releases will specifically be targeted for you to confirm that the final content has been inserted. Remember that SonicSpider is NOT a copy editor, if there are spelling or grammatical errors it is your responsibility to note them and send the corrected text. The minimum unit of replacement is the paragraph.
3. **Release Candidates** – These are "feature complete" releases, where any one of the releases may be deemed "finished". There will be ANY number of Release Candidates. The goal at this point is to get the project delivered,therefore it is important to provide complete and final feedback. Don't address single issues and do a complete review and provide a complete and "final" corrections list.

Following are some of the possible release scenarios:

- Prototypes of home and a sample interior page. This release focuses on layout issues. Text is just a place holder. Images, spacing, colors and the visual layout of text is to be reviewed.
- Preliminary working models of individual forms/pages (Betas). These will be released one at a time (or in logical groups) and will contain about 50% to 80% of the designed functionality.
- Content insertion – This release focuses on the real content. Review the

content for spelling and grammatical errors. Send back full paragraphs.

- Integration of forms/pages. Once the primary functionality of the user interface is complete the individual forms or web pages are completely integrated and tested as a whole. At this point you should be looking at the site or application as a whole. Review the flow of pages, menus and any other navigation issues.
- Dynamic or Data entry modules. Review the specification and insure that all functions outline there have been addressed. Attempt to “break” it by doing unorthodox sequences or inputs to review how it responds. The application should detect problems and respond or degrade in a meaningful manner.

The sequence of releases in each category and what they address can repeat a number of times on different portions of a project. Therefore it is not a linear process but a "feedback loop" process that involves the close teamwork of SonicSpider development staff and you, the ultimate users of the application.

A Final Word

It is hoped that this document helps the reader understand the process required for a successful project. It has been documented in many articles and books that “on the average” only about 10% of all projects are “successful”. When studying the projects that succeeded vs. those that did not, it is clear that most of the successes were those that adhered to the contents of this guideline.

This process may seem complicated at first glance, but after over 20 years of experience in application development, this guideline fairly represents all of the components that would be required to successfully complete a project.   Anytime some part of this process was side tracked, short changed, or omitted, it has often spelled the doom of that project. The SonicSpider team can ONLY assure you that your project will be successful IF this process is followed in a manner that is appropriate to the size and scope of your project, of which the SonicSpider team reserves the right to determine the appropriate level of adherence for your project. If you chose to take short cuts, SonicSpider can not be held accountable.