

## QA?

by Chris, July 2008

According to [everyone's favorite Project Management glossary](#), QA, or Quality Assurance, is "A planned and systematic pattern of all actions necessary to provide adequate confidence that the item or product conforms to established technical requirements." In other words, QA in the Newfangled project world is a plan to make sure that everything works as it was intended to work and looks as it was intended to look.



We recently revamped our [Project Anatomy document](#), which specifies all the steps and roles involved in a typical web development project. We use it to properly plan all of our projects at the outset, making sure we've allocated the appropriate time and resources in order to meet everyone's expectations, as well as to track the progress of a project along the way. In fact, we've even set up an online version of the Project Anatomy that our Project Managers can use to verify that each necessary review, approval process, etc. has been completed before a project moves to a new phase. If you're looking at the document now, you'll notice that there are three specific steps dedicated to QA. The first two occur during the programming phase of the project, and the third occurs immediately after a site goes live. These QA steps are guided by involvement with almost everyone on the team, requiring a thorough review of every aspect of the site's design and functionality and producing a detailed report that we use to make any necessary corrections. However, there are two other QA steps in our Anatomy that aren't as immediately obvious, but are probably the most important.

### Code Review

After the second round of QA is complete, our engineering team performs a rigorous code review with the developer on the project. Often some element of functionality could be programmed several different ways, and it is critical that the best way is chosen to ensure that the functionality scales well as the site grows and is used. If this doesn't happen, a site could experience all sorts of bugs or unpredictable behavior as time goes on. The purpose of the code review is to ensure that our programming meets the standards of the Engineering department and will not become a liability later. This is also a great way for our CTO and Lead Systems Engineer to continue to mentor our developers.

### Integration is QA

Once the programming has been completed and reviewed, the site is ready for content entry. This step can be found under the heading of "Integration." While there is no specific QA task listed under this phase, I believe that content entry is one of the most effective and important QA efforts for any project. Typically, this is the point in a project when a client is able to fully experience the reality of their site or application for the first time. While they have worked closely with our team on prototyping and designing the site, the process of using the content management system, creating, and entering content is when all the "dots" are connected and made real, and often the first point at which expectations are clarified. You see, no matter how thorough a prototype is, sometimes there are concepts or needs that cannot be communicated until you are immersed in an actual working and producing environment. This is similar to the "blank-slate-shopper" phenomenon: Have you ever seen a review of a book and thought that you'd like to purchase it, only to find that the next time you are actually in a bookstore you have no idea what you want or where to start? This is because we tend towards reactive rather than proactive thinking. We hear about something and react to it with, "Yes, I'd like to read that," yet when we get to the store and are surrounded by thousands of books, we react to them all by losing focus. While we find the prototyping phase, being a proactive step, to be extremely effective and critical to our process, we use QA steps to catch any ramifications of reactive thinking during a project and know that the process of content entry will do the same.

As I've written before, QA does not ensure that a project will be 100% bug free. While some bugs are due to sloppiness or haste and can

be prevented by thorough QA steps, others are the result of unforeseen functionality conflicts that may not become evident until a site is being used, despite the best intentions and foresight of the programmers. As with any development project, bugs like these should be expected and encountered with patience (this goes for us just as much as our clients). While we hope that our many stages of QA will mitigate the frequency of any bugs occurring, we are definitely not surprised when they show up.

Ultimately QA is a collaborative effort. We've got plenty of measures to confidently assure quality, but we also depend upon our clients to respond to each step along the way and make sure their expectations are clearly communicated. We work towards having a great working relationship with our clients so that we can foster an environment where that communication is the foundation of real quality assurance.