DOING IT RIGHT THE SECOND TIME….

PROTOTYPING WEB APPLICATION

USING DREAMWEAVER

Gauri Ghare

eTransitions, Inc.

Have you been wondering about the title, thinking that if she is not doing the job right the first time what is she so proud about and even talking about it as a great achievement? Our industry motto has always been

If you fail for the first time, hide the evidence.

and here I am talking about it openly.

WHAT HAPPENED THE FIRST TIME?

For a minute, think about the time we spend in developing web application for our client. We spend lot of time to create beautiful layouts, graphics and then spend several days to write code and test it. All this to do it right the first time. But, the users are new to the web application technology and they do not know how the application should look and function. Sometimes, they have been using a particular business option for such a long time that they forget to describe an important step accurately and at the time of implementation, point out several instances when a particular thing is not what they want. The navigation that we designed was beautiful but the users are confused. Result? We throw away most of the code and start again.

WHAT IS RIGHT?

Simply put: When the application does what the user needs, in a way that is easy for them to understand then, it is right. Note that, there are many right solutions. We are not fussy. We want the one that is quickest, easiest and cheapest to develop.

WHAT ARE WE NOT DOING RIGHT?

You name it, the page layout, the graphics, the navigation, it could be any one of these or all of these. Remember how picky and choosy are our clients most of the time. Result? We have to throw away most of our layout designs, graphics, navigation charts and start again.

DO ALL OF THE PIECES FIT TOGETHER?

Most developments consist of custom code, some pre-existing components and /or some third party components. We think they will fit together and the application will work. But as the development proceeds, we find an unexpected behavior. Now we have to find a workaround. We may have to discard a particular software component and find a replacement. Sometimes we may have to write our own piece of code to do the job. Result? We throw away most of our code we worked on till that time and start again.

Conclusion? We did not do it right the first time, did we?

Since, it is obvious that we do a lot of re-work anyway, why not start with a prototype that lets us figure out how to do it right may be the second time. Otherwise, we will not really know how many times we will need to rework, before we get it right.

WHAT IS A PROTOTYPE?

A Prototype is a model to demonstrate specific functionality of the application we want to develop.

We inherit the concept from the engineering field. Structural Engineers create models of buildings, bridges and townships. Chemical Engineers have pilot manufacturing plants. In Software development, we use prototypes to model functionality of the software.

WHY SHOULD WE DEVELOP A PROTOTYPE?

- To demonstrate or highlight functionality of the application

  We develop a prototype when we want to highlight important functions of an application We can use it to confirm user requirements of one or more important parts of the application feature, functionality or design. A point to remember here is that a prototype is demonstration of only a part the application or function. It is developed with this thought that we may have to redesign part or the complete function.

- To demonstrate to our client how the technology works.

  Our client may not be able to visualize how the technology is going to work to produce the required result. A prototype helps to demonstrate how a particular functionality works. It is worth the efforts to create a prototype to demonstrate functionality of the technology, gain user confidence and approval for the technology. Looking at the prototype, client will understand what they will get at the end of development and whether or not it meets the company laid out policies and goals.

- To test a particular piece of technology and its behavior:

  If we want to test the technology to confirm that it works exactly the way we want, it is a good idea to develop a prototype. This will help to confirm the way the technology works. It will also help to identify a particular behavior or limitation, any other issue that may cause concern during the development. The prototype will help to identify the different modules of software that will go together to give closer perspective of the software. It will help to make decision about using a particular technology, its pro's and con's and also the performance issues.

- To get user feedback

  We can also use it to demonstrate the required functionality and get user feedback. A prototype can give our clients *the user experience* of the application. Working with a prototype is similar to using the actual application with only a particular functionality. So, the user responses received during the demonstration are very important and useful. A prototype can also help to prepare users for new application long before the application is ready for implementation.

HOW TO DEVELOP A PROTOTYPE

The important point in developing prototype is that, we should be able to develop and also modify it quickly and easily. With this in mind we must make use of available tools and techniques.

THE TOOLS

- Quick and Beautiful

  We choose tools that will help us to create layout and graphics easily and quickly. We should be able to use the tools without any formal training. Formal training involves time and skills. If the tool is intuitive to use then formal training is not required. Of course, a lot of times good reference books help. We can make use of such material to achieve what we are looking for. Another important point is that there should be no coding involved. We don't want to spend time in debugging and re-coding. There is a possibility that we may have to throw away part or all of the code. Therefore, we do not want to spend extra time or efforts on this step than required. I call it *Quick and Beautiful!*

- Easy modifications

  The tool we use must allow quick modifications to our prototype. Again, we do not want to spend time in testing, coding and debugging. After all it is just a Prototype and not the final product!

- Use of the prototype as part of the final product

It would be very nice if we could use at least some components of the prototype as part of our final product. After all we did spend some time creating it! Utilizing components of prototype as part of final product can be achieved with a good Web Authoring tools and careful design. This will help us to reduce time during actual development.

THE TECHNIQUE

- Storyboarding and Navigation Charts

  The storyboarding technique outlines the web application navigation strategy. A detailed diagram of the navigation strategy points out if we have missed any important displays during the web application. It also shows how the navigation from one page to another is carried out. The page layout covers the details like where is what displayed on a page. Things that are marked on a typical layout page are the logos, the toolbar, the frames, the static information and the displayed data samples. The page layout strategy and the Navigation Charts help to give our programmers and us a graphic view of the application's logic.

- Creating Page Layouts

  We use graphic tools and suitable authoring software to create the page layouts and navigation charts. Page layouts are the graphic representation of how the pages will look during the application execution. These are also the representation of actual pages that will be created during the application development during detail design phase. We usually have one page layout for one module or function of the application. Each page should include representation of use of frames or tables, logo display, toolbar display and areas where static information and dynamic information will be displayed. The page layout should include the name or title of the page representing the function it will execute.

  These pages can be grouped into different categories. These categories are:

  1. General information pages which include log-in / log-out page and a home page,
  2. Query selection pages
  3. Result display pages
  4. Message pages

  There can be subcategories in each of these categories. (e.g. Message Pages may have subcategories like confirmation messages, Warnings and Alert messages and Error messages)

  When we do the layout design of each page, we can easily identify the static objects and the dynamic objects on different pages. After we have categorized our pages, we design a layout and create a template for each category and subcategory identifying the static and dynamic objects

- Creating Templates

  This technique will help us to apply changes easily and uniformly on different pages and get a consistent look for our application. Templates are like boilerplates for a set of pages. During the Page layout design we identify which of the objects appear on different pages under a given category. These objects are static objects for such pages. For example, a company logo or buttons on a toolbar are static objects. Similarly we identify the objects which change on every page. These are the dynamic objects. For example, the data retrieved from a database changes and therefore it will be a dynamic object. The area or region on the web page template where static objects are placed is called as non-editable region of the template and the area where dynamic objects are placed is called the editable region of the template. We can create a new page based on a specific template since the static objects are readily available. Now, if we make changes to the template then all the pages that are based on a particular template can inherit the *new look*.

- Creating Sample database

  We should make use of a sample database. It will help to get sample data from existing database required for the functions we are working on. It helps users to relate to the information that is displayed against the different prompts on the screen. It makes them feel more comfortable to see

data that they recognize as meaningful. At this time they can even point out whether or not a function is useful and is actually giving them the expected output.

- Writing Query statements

  We then write and incorporate the query statements that retrieve the data from our sample database. This should be done for each page and each dynamic component we identify.

- Putting It Together

  When we put all these things together, our prototype is ready for demonstration and user critique of the functions we have put together. Don't get upset with user criticism because that is important feed back for us and we can either modify or change functionality of the application or module. We did develop the prototype to get user feedback and we did not expect it at all to be right the first time.

- Getting feedback from globally dispersed users

  Once the prototype is ready we can put it on a Web Server to which our intended users have access. We can email the URL to the users. They can look at the application, use it and give us the feedback we are looking for. Since, it is intended for web use, getting users to use it on the web is a good idea. This also solves the issue of getting feedback from globally dispersed users.

Now, all things in the prototype are on a small-scale basis including the size and volume of the database. It is limited to a specific part of the final product. Therefore, the prototype does not take long to develop. We can apply the same prototyping technique while developing other pieces of our application and achieve faster, quicker and consistent development.

During the presentation there will be a step by step demonstration on how to create templates and how to use them using DreamWeaver 3.0 and ColdFusion 4.5.

SUMMARY

Prototype helps you get development finished quickly, easily and cheaper.

We use prototype to confirm user requirements, design and functionality of application and use of technology.

Templates will help you to get a consistent looking application, which satisfies user.

Use of good tools will ensure that changes can be applied easily to the application so get the prototype done faster.

About the Author:

Gauri Ghare has over 10 years of Analysis, Design and programming experience in various languages and databases using different CASE tools and project management tools. She has worked as a project manager on different projects including web related projects under the banner of eTransitions, Inc. Working closely with users she has gained valuable experience in interface design for web environment. She may be contacted at gghare@etransitions.com.