The System Development Life Cycle (SDLC) of Web Development

By Don Demrow - © 2007, Don Demrow

Problem
The design / development "process" that is in place at many organizations is no process at all, which results in multiple costly revisions, and a tendency toward non-adherence to any established standards or conventions, if in fact, such standards and conventions even exist.

Objective
To adopt a process of iterative "pigtail" progression to accomplish expedient, efficient design and development on future projects. This refers to a cyclical revision process contained inside a linear progression, illustrated as a curly line, or "pigtail."

This process will facilitate transparent communication to establish effective interaction and dialogue between all parties involved in the conception, design, development, approval, testing, implementation, and revision process of websites and applications.

Solution
Establishing a systematic approach for design and development of new websites and online applications will save time and money. Applying a **System Development Life Cycle** model is the most efficient way to do this.

If roles, standards, objectives, and expectations are clearly defined through the process, from beginning to end, there can be little confusion as to who was supposed to accomplish a given task.

We must define the following questions prior to any design or development:

Who – Who will design, develop, approve, test, launch, review / revise.
What – Scope. What exactly do we need to accomplish? What is the ROI?
Where – Will this be an internal tool, or a public-facing B-to-B application or site?
When – Reasonable deadlines.
Why – Why build it? Will this site or application save time / money? Generate leads?
How – Implementation and presentation – What is appropriate? Will we do this with Server-side code? Static HTML? Flash? How will it look? ASG Corporate? Modern? Business-like?

Definition of Roles
The "**Who**" part of the process would be designated by the managers, or leaders of the design and web teams. (Example: *"Matthew will design the flash introduction, Mark will design the page layout, and Luke will cut graphics, and code the application."*)

Likely, the initiator of a project would define **What**, **Where** and **When**. (Example: *"We would like a lead automation system, by end off fiscal 2007. It will be an internal tool, but with a public-facing component, which will be the lead form. We wish to capture leads and get them into our existing AS400 database."*)

Why would be a brief explanation of the business objective, and any business pains remedied. (Example: *"It takes hundreds of hours to enter leads manually, and it is too slow. By the time we contact everyone, they have already purchased software from a competitor."*)

How Technology: defined by web team, presentation defined by design team and web team. (Example: *"Flash for introduction, ASP or coldFusion with a SQL Server Database."*)

Website / Application Requirements Outline

1. **Analysis / Requirements-Gathering**
   A. Define Overall scope definition
   > (Examples: *"We need an online application that will allow users to lookup their order status, and add to their order,"* or *"We need a website for our new product family."*)

   > This stage is NOT an open forum for users to become designers. However, "Storyboard" or "Cocktail napkin" preliminary designs should be welcomed at this stage.

   > During this stage, parties requesting the website or web application should be solicited for information about the business objective they are trying to achieve.

   > Input should be restricted to a formal Site/Application Requirement Form. (Preferably, online)

   B. Define Sign-off Authority / (Final Approval Point-of-Contact)
   > It is essential that the final "sign-off" authority be named in the requirement specification document.

2. **Design** – This begins the iterative process of:
   A. Review – <mark>Is it graphically appealing, and appropriate for the intended purpose?</mark>
   B. Revision

3. **Design Approval**
   > At this point, the design is locked. "Scope creep" is very costly in terms of time and money. At this point, the specification is **locked** – additional features will be addressed in future revisions. This is necessary to keep development time and cost on track.

4. **Development**
   A. Review – <mark>Does it satisfy business objectives and requirements?</mark>
   B. Revision

5. **Prototype Test / Approval**
   A. Review – Final management review prior to launch. <mark>Does it perform to specifications?</mark>
   B. Revision – Final revision for this version.

6. **Launch** – The site or application goes live. (Final approved version, set in step #3, and any last-minute, "finishing touches" were applied in step #5.)

7. **Review / Revise** – an iterative process – can be thought of as a sub-cycle through the SDLC. Notice that this is much the same as items 2 and 4, but applies to the entire process, rather than just one aspect of the SDLC.

Lather, rinse, repeat.

End of this **version**, beginning of new **version**.

Review / Revision Cycle

```
        ┌─────────────┐
        │ Start Cycle │
        └─────────────┘
               │
               ▼
   ┌──────────┐      ┌──────────┐
   │  Review  │◄─────│  Revise  │
   └──────────┘      └──────────┘
        │                  ▲
        ▼                  │
   ╱──────────────╲   Yes  │
  ╱ Changes Needed? ╲──────┘
   ╲──────────────╱
        │
        │ No
        ▼
  ┌────────────────┐
  │ End - Next Step│
  └────────────────┘
```