STRATEGIC WEBSITE TECHNOLOGIES

(WEB SITE MANAGEMENT)

Upgrade or Start Over

Some people think that scratching their old web-site and starting a brand new one is easier than trying to upgrade the web-site.

Don’t forget that

- No web-site is ever done.
- All sites need attention all the time.
- Every year new tools and new technologies appear. So if you start rebuilding you will need to rebuild again.
- If you plan for maintenance, you can prevent having to rebuild your side over and over.

Motto: PLAN FOR THE FUTURE

Planning Ahead

When you are on a tight deadline there is the temptation to begin coding immediately.

- However, if you plan ahead you might save yourself several rounds of bug testing.
- Your team members will have a better idea what the finished product will look like, and work accordingly.
- Even establishing your goals and determining the things to be done to accomplish these goals is better than not planning at all.

Motto: PLAN BEFORE YOU START CODING

Maintaining Your Website

You maintain your website to ensure that it performs well.

Continuous maintenance of a website will help you

- to spot troubles while they are considerably small
- to make the (inevitable) upgrade much easily
- to reduce your workload in the long term
- to produce list of identified site production tasks and a history over the lifespan of the site. (This can be very useful for new employees).

Motto: DON’T LEAVE MAINTENANCE TO THE LAST MINUTE

What You Should Maintain Regularly

- Performance of every page on your site
  - Content
  - Hyperlinks
  - Formatting
- Navigation Bars
- Legal notices (copyright) and points of contact information
- Backend
  - unused code and graphics
  - paths are correct and consistent
  - total file size of pages as they load
  - formats of files
- Performance on different or new browsers

Upgrading Your Website

Sometimes, performing measures to maintain the code is more trouble than simply rewriting the code to perform more functions in fewer lines.

3 kinds of upgrade:

1. Backend upgrade (to make production run more smoothly, reduce chance of redundant files).

2.Code upgrade (Changing HTML and scripts to comply with latest specifications and to work better on different platforms etc).

3.Theme upgrade (Changing the look and feel of the site).

* An upgrade is a step above and beyond site maintenance.

When it is time to upgrade?

There are a few signs that will tell you it’s time to upgrade:

- Your primary competitor overhauled theirs and it outperforms yours. (Externally driven)
- You have added enough content to reorganize the site’s content structure.
- Deciding to adopt a new HTML standard or backend technology.
- Adding a new type of functionality (such as a search engine)
- Realizing that the code you’re using to maintain the site has become very messy.

THE KEYWORD IS UPGRADE **NOT** REWRITE

Implementing Maintenance

A regular site maintenance regime requires:

- Time
- Money
- Personnel
  - Team work
  - Documentation
  - Lack of personnel or lack of people who developed the site initially could be a big problem.
  - Put maintenance and documentation into a contractor’s work agreement (for third party development).
- Organizational Support (Managers and developers should speak the same language).

The Idea Questions

These questions will help you to determine every possible reason you're building your web site.

- *Why is my organization maintaining or upgrading a website?* (Keep up the profits, give info to the public)
- *What is the purpose of my site?*
  - to inform (presume prior interest and knowledge - Academic web sites)
  - to teach (a specific topic)
  - to guide (task oriented, step by step instructions)
  - to act as a resource (organized data to allow a user to find specific information)

(A site can fulfil more than one informational strategy.)

- *What will my site offer that a product based in another medium wouldn't?* (brochureware and shovelware websites)
- *What would I like to see as my ideal website?* (without considering the sales and human resource considerations)
- *What is the minimum acceptable standard I can get away with?* (It is impossible to make everyone happy)
- *What sort of content boundaries do I have?* (Last minute changes in the content are unhealthy) (Draw up milestones and stick to them)
- *Do I know whom I'm building the site for and what my user is like?* (market research, visit competitors websites, can you reach the newly emerging markets)
- What kind of interaction? (forms, e-mail etc. e.g. university students interested in travel spend 20hrs/week online and have low-end browsers)

The Implementation Questions

These questions will help you impose order and control on your website development process.

- *Why am I the one doing the work?* (experience etc.)
  - What skills and technical experience does your team members have?
  - Do you need extra knowledge (people) to accomplish the work? (Be realistic about learning curves and deadlines)
- *What sort of website am I starting out with?*
  - Starting with a website could be good or bad.
  - How have the site workers treated files as they get older?
  - Do you have a clearly defined archiving system set up?

- *How much time can I allot to the website*?
  - *What skills does each team member have?* (Do you have a backup person for each skill)
  - *How much time does each person have to devote to the project?*
  - *How much time would integrating the work take?*
  - *Have you allotted time for work review?*
  - *Can you live up, if a team member leaves the project?*
- *How will I handle people leaving? What redundancy have I built into the production process?* (who's doing what? Who's assisting who? Is everybody happy and can use their skills?)
- *What do my nonhuman resources look like?* (Servers, rooms, software etc.)

- How much maintenance will the site need after it is completed?

(Depends on the purpose of the site (news-site vs trade-show site)

- How much backend maintenance and upgrading am I willing to do?
  - Are you going to carry content to archive files in a regular basis?
  - Are you going to integrate new technologies such as SSI into your website?
- How do I know I'm done?

(Very few websites are ever done)

(The real question is, how do you know when you are done with the backend?)

The Technology Questions

These questions help you to assess what tools you need to develop your site, and to keep your technical staff happy.

- *Should I build the tools I need or should I buy them?* Consider
  - **Return on investment,**
  - **support,** (The tools you write may have better support)
  - **extensibility,**
  - **time** (consider also the testing time)
- *What do I want the tool to do?* (A good tool should be like a toothbrush)
- *Is the tool cross-platform?* (runs on different platforms)
- *Is the tool producing reliable results across browsers?*

- *How important are fancy futures?* (Determine where to draw the line)
- *Can I get at the source code for the tool?* (Apache server)
- *Can I modify or extend the tools functionality?*
- *How important is standard technology to me?* (e.g dynamic HTML – Do you have a backup plan, if you don't follow standards)
- *How much support is this tool going to have?* (Don't forget that in-house development is no guarantee of in-house support).
- *How will the tool affect my overall production deadlines*?
  - *Does the tool brings any obstacles in production?*
  - *Does it impose a new way for the staff to work?*
  - *Does it impose limitations on the code or scripts?*

The tools you will need

A telnet application: Should provide compatible terminals with your site.

A drag-and-drop FTP application: CuteFTP, FTPExplorer.

An HTML Editor: Providing color tags, if possible. Homesite, HotDog, Macromedia DreamWeave, MS FrontPage, MS InterDev

Graphics Editors: Corel Photopaint, Adobe Photopaint, GIF Animator, Macromedia Fireworks, Flash etc.

Multiple Browsers: Latest versions of IE and Navigator plus if possible a few old versions.

THE DEVELOPMENT PROCESS

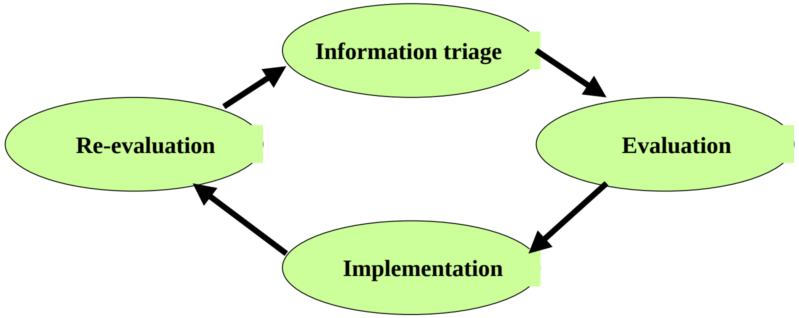
Conventional software development consists of 6 stages:

1. Conception
2. Design
3. Implementation
4. Testing
5. Documentation
6. Release

These steps are done in a strict order and not repeated (until the next version).

Website development, however, is different.

There is no concrete 6 step development cycle.



Information triage

Re-evaluation

Evaluation

Implementation

The Web Development Cycle

Information Triage

The news is flowing from everywhere in the information age.

You should develop a set of criteria for the news you receive and select you news resources accordingly.

4-5 news resources and mailing lists is better than trying to handle 10-20.

There is a difference between data, information and knowledge.

Data: Composed of raw facts or observations

Information: Data applied to a particular context or idea

Knowledge: Being able to apply information to a specific situation.

To collect information for your site you may use the following ideas

1. Determine the information relevance
2. Develop an expiration date for relevant information (prioritise what to read – you don’t need to read everything at the moment)
3. Set topical parameters (You can’t know a lot about everything)
4. Use news digests
5. Swap information with your colleagues

Evaluating Information

The information you gained include

- News of new technologies
- Answers to the questions we have been asking
- Individual observations and experiences

The purpose of evaluating information is to figure out how to apply everything you learned to the development process of your site.

You should consider **feasibility** (is it possible to implement these given the fact that avg. customer has 14.4 modem) and **time-frame** (when do you need to apply the new info to your site) in your evaluation.

Implementation

This step involves the application of the evaluated information to your site.

This includes

- Building a better backend
- Writing faster, cleaner, more flexible HTML
- Migrating from a static website to a dynamic data-driven one (One page at a time or mix and match assembly).
- Optimising graphics and animation
- Using scripts to perform data transactions
- Building tools to make your job easier

Re-evaluating Your Work

This job is crucial but easily forgotten. Can be done in two steps:

A week after launching the site:

- What was the most useful information that passed the evaluation step?
- What you wish you had known?
- What did you not find useful at all?

A month after launching the site:

Here you try to determine why of what-why-how have been met.

- Is the site working better, or did it break more things than it fixed?
- Did the site repairs have decreased the maintenance work?
- Are you happy with the final product?

You should start brainstorming what the next steps are toward beginning the cycle again!