



Using a User Centered Approach to a CMS Adoption: Planning, Process, and Implementation

Session handout for NERCOMP annual conference
March 20, 2007

Project website
<http://go.brandeis.edu/cms-project>

Includes project reports, documents, worksheets, and more.

Start to finish

About Brandeis

- Established in 1947
- Self-described as “a premiere small liberal arts research university”
- 5000 students (undergraduates & graduates)
- 4 schools (Arts and Sciences, the Heller School for Social Policy and Management, the International Business School, the Rabb School of Continuing Studies)
- 33 centers & institutes

About the Brandeis web

- 40,000 HTML pages on www.brandeis.edu
- About 4000 PHP pages
- Websites for schools, programs, and departments
- About 250 web publishers

About the Brandeis CMS

- Selected Hannon Hill’s Cascade Server; contract signed early May 2006
- Conducted pilots during Summer-Fall 2006
- Now 20 sites in the CMS; 65 web publishers
- Conversion of large Library & Technology Services site ongoing
 - 6000 pages, 50 web publishers

Part 1: Define the problem.

Step 1. Brainstorm!

Tips:

- **Don't go it alone.** Include your entire web team in this exercise, and consider bringing in some other folks from your organization to provide additional perspectives. Making this brainstorming a group effort will allow you to gather a broader range of ideas than you otherwise might have. This will take more time than you expected, but your investment in time will pay dividends as the process unfolds.
- **Clearly define the problem.** Focus on identifying areas in need of improvement & avoid the blame game.
- **Be honest. Being uncomfortable is OK.** Take a hard look at the state of your institution's web and the state of your support services. This can be an uncomfortable conversation, but it exposes critical vulnerabilities & deficiencies.

Outcome: Vision & Scope document. Included:

- a vision statement describing how this new system would transform web publishing at Brandeis
- functionality the CMS would provide
- assumptions & dependencies
- preliminary definition of project scope.

Step 2. Learn about your users and their needs.

Tips:

- **Think clearly about what you want to learn.** It's tempting to dive into user research with only some vague ideas about what you need to learn. However, taking the time to think clearly about what you need to learn from your web publishers is the only way to gather data that'll be useful for your process. Your constituents will appreciate your thoughtful approach, and the results of your research will probably illuminate issues that didn't arise in your brainstorming.
- **Avoid being constrained by your assumptions.** Our brainstorming process helped us better understand our web environment and pull together group knowledge about our how web publishing works at Brandeis. However, we needed to take care that the knowledge gained through this process didn't produce assumptions that blinded us to the new and unexpected learning emerging from our user research. Be aware of your assumptions, and try to put them aside as you consider the results of your user research.
- **Spend some time learning how to conduct user research.** User research is a discipline. While we're not experts, we put lots of time and effort into learning some techniques for conducting effective focus groups and into planning our research methods.

- **Not every user research method works in every situation.** We used our focus groups effectively to further explore ideas raised in our brainstorming and define the web publishing problem from the perspective of the web publishers. However, we thought our follow-up survey would broaden our reach and bring in new ideas that the focus groups hadn't unearthed. Instead, the survey mostly repeated ideas that we'd already heard. A better approach would have been to use the survey to gather additional ideas that hadn't emerged in our brainstorming, and then use the focus groups to examine these ideas in more depth.
- **Take your time.** We spent literally weeks analyzing the data we gathered, considering their implications, and writing a cogent User Needs Summary. That was time very, very well spent – our conclusions informed the rest of our process.
- *Outcome: User Needs Summary document.* Includes:
 - Project introduction
 - Description of research methods
 - Findings and conclusions
 - Recommendations
- *Helpful resources:*
 - *Focus groups : a practical guide for applied research.* Krueger, Richard A. Thousand Oaks, Calif. : Sage Publications, 1994.
 - *Understanding your users a practical guide to user requirements, methods, tools, and techniques.* Courage, Catherine.; Baxter, Kathy. Amsterdam ; Boston : Morgan Kaufmann Publishers, 2005.

Part 2. Define the solution.

Step 3. Develop business requirements.

Tips:

- **Maintain a high-level view of the problem.** Resist the temptation to bundle functional requirements into your business requirements document. This will only cloud your perspective on the problem you're trying to solve.
- **Develop user scenarios.** Although we planned to develop user scenarios as part of the business requirements analysis, we didn't. We later developed a great set of user scenarios, informed by the User Needs Summary, that helped us effectively describe to our constituencies how the CMS would help them publish more effectively. However, developing those scenarios earlier would have helped us think through the business requirements more carefully and would have made its recommendations more concrete.
- **Use the BRD as a blueprint, not a project plan.** While the Business Requirements Document informed our work throughout the rest of the process and provided a roadmap for the project, we occasionally diverged from the course we'd set. Use the document to light your way toward a solution, but don't treat it as a binding contract.

- *Outcome:* **Business Requirements Document.** Includes:
 - Statement summarizing the problem to be solved
 - Project goals and objectives
 - Description of current state, processes, and workflow
 - High-level requirements for a solution
 - Assumptions and dependencies
 - Preliminary project timeline

Step 4. Develop detailed functional and technical requirements.

Tips:

- **Get participation and feedback from all levels.**
- **Formal documentation is valuable now & in future.**
- **In requirements gathering sessions, seek equal airtime.** The goal of Be prepared to moderate and steer the meeting, but let it be free. Want ideas from all (make sure quiet people can talk).
- **Don't let the technical team, simply because they are busy, get off the hook from contributing.** However, don't be afraid to be lighter on the process with technical group: they work doing this sort of evaluation all the time (also, technical considerations may be more absolute, and thus require less process to define).
- *Outcome:* **Functional and Technical Requirements Documents.** Includes:
 - what must the system do in order to improve web editing process
 - rankings of feature importance based on end-user needs
 - grouping of items from essential to desired to extras
 - checklists that can be used during product demonstrations

Part 3. Apply the definitions to answer the need.

Step 5. Evaluate products on your short list.

Tips:

- **Beware process fatigue.** After all these steps, the evaluation battery can seem like a final demand of ‘too much time’. However, it is very important in this step to go in-depth and have all-hands participation: this is where the hard work gets applied and ‘becomes real’. Be sensitive to people’s needs, but encourage participation, as it is essential to buy-in.
- **Gather feedback at every opportunity.** During vendor demos we provided index cards to have people write down additional questions. We collected them and contacted the vendor to seek answers as post-demo follow ups. At the end of each presentation, each team member was given a worksheet with two pages of most desired requirements. These rankings, and especially the freeform comments, factored highly in final selection.
- **Complete your due diligence even if a ‘presumed winner’ emerges.** During the demonstrations, it may be that you’ll perceive a product to far surpass all others. Don’t let this leader derail you from documenting each product evaluation – someday questions will be asked about why you didn’t like product X by company Y, and having the documentation for reference will prove valuable.
- *Outcome:* **Final product selection**

Step 6. Implementation

Tips:

- **Have a vision.** CMSs don’t come with vision – that’s on you! Use your organization’s collective intelligence and competing priorities to forge a vision for deployment – ask not only, “what do we want from our CMS?” but also “what do we want from our web presence?”. Leverage Comm, IT, web, and academy resources to form a vision that’s flexible enough to handle shakeup in the queue, accommodating to the academic calendar, and turns a phased effort into a successful and predictable whole.
- **Learn from your pilots.** After we purchased our CMS, we thought carefully about the size, scope, and variety of sites that we wanted to work with and the learning outcomes we hoped to achieve. From small department sites to a fully CMS-built undergrad admissions site, we sought to test our assumptions. Taken seriously as a deliberate pilot phase and not simply as the first sites in the queue allowed us test various hypotheses and remain objective in what we learned. Your reward is a challenge – many folks watching our progress lined up to say, “I want that too.”

- **Review People, Process & Policy.** Your CMS deployment will absolutely force an assessment of these three, and may shine an uncomfortable light on the current state of your process, people and policies (or the fact that some don't exist). While dovetailing with vision and EOS, I put this here as a portent – you will find that these three resources will all come under review when you begin thinking about life with a CMS. Your ability to be a diplomat for what is possible (and required) will make these discussions easier.
- **CMS is not a Panacea:** I've yet to find a fix in a box, and a CMS is no different. Things will be more difficult at first. It is normal (quite likely in fact) to think that a CMS will dramatically change the way many people work – and the way everyone working on the web works. However, the curve comes up fast and turns quickly and easily – that is, when things start to settle in, users can be turned on and trained very quickly. In many ways, the user needs analysis and vendor selection is very much the easy part.
- *Outcome:* **Deployment strategy**