Deployment Panel:
Planning and Implementing for the Big Day

Notes from CAMP Directory Workshop
February 3-6, 2004 Tempe Arizona

Moderated by: Mike Conlon
Panel members: Daniel Arrasjid, Robert Banz, Mike Berman, Tom Jordan
Scribe: Nate Klingenstein

You cannot overestimate the number of people you need to be able to communicate with during infrastructure projects if only to keep some from becoming adversaries at a later point.

Implementation day can be very busy, and it is not the time to be ironing out communication plans. The campus needs to understand an implementation is going live and there need to be effective communications in place. Training should be thoroughly completed and effective feedback channels in place, allowing you to find issues early and isolate whether they're systemic or sporadic. Complaints are an extremely useful source of information on where infrastructure needs improvement.

Communications to technical staff are different from communications to everyone else. The real communication error UMBC made in deployment of its enterprise directory was failing to inform the technical folks sufficiently of its full capabilities and contents. This meant developers and applications often failed to make sufficient use of the directory.

It's possible to communicate too early, while things are still being figured out; this can cause people on campus to panic. If it's impossible to answer more specific questions about launch dates or the impact on the community, it may be better to wait.

Start with conceptual communication, encouraging people to think about an enterprise directory in conjunction with the enterprise portal. Over the course of the year, this was transitioned from an education and sales pitch presentation to information about the project status as details became more concrete.

An audience member suggests a different deployment strategy: there should be no big day. There will be many connectors and bridge systems in place to gradually migrate from old systems to newer systems and avoid single big switch-overs. However, the amount of time it may have taken to perform a sequential transition may not be worth it relative to the single downtime.

Time spent on GUI and front-end design can often greatly reduce the amount of training and user education needed; however, for some things that can't be simplified, such as LDAP schema, training may be the only viable approach. People frequently learn best in context. Training has a too-soon too-late aspect too: if people are trained on mock-ups or incomplete systems, it is very difficult for lessons to be thorough and understandable;
however, if they're trained too late, there may be too many people to train in too short a time.

Back-out and contingency plans must be in place in case the deployment isn't going well enough or performing well. On deployment day, it's relatively straightforward to revert to the previous system by turning the new one off and the old one back on; however, after a couple days and accepting a few changes into the new system, if the need to back out occurs, it becomes much more involved and being prepared is invaluable.

One of the most important things after going live is continuing discussions between help desk staff and technical staff. Eventually, there may be good pattern-recognition tools on the help desk, but for now, a knowledgeable person in the trenches has to find trends and work with a technical lead to identify underlying problems and root causes. There is no substitute for this invaluable part of the launch process.