

Cutting Through the Clutter: What Makes an Intranet Successful?

Eleven years of portal development at the Wharton School yields successful strategies for enterprise information management

By **Kendall Whitehouse**

Most large enterprises now have some type of portal to organize their internal information services. Few, however, have the long history of the SPIKE project at the University of Pennsylvania's Wharton School. Originally developed during the 1994–95 school year, SPIKE was an intranet information portal before the term “information portal” became common, and it was one of the first enterprise information systems in higher education.¹ Updated each school year in response to feedback from current students, Wharton's SPIKE has expanded beyond the Web to include everything from large video screens throughout the Wharton campus to students' handheld devices. Through more than a decade of development, the school has learned a number of lessons about what makes a successful intranet information service.

A Brief History of SPIKE

The SPIKE project began in 1994 when a group of Wharton MBA students approached the school's IT organization about revamping the school's information environment. At that time Wharton provided the usual set of electronic information services, including e-mail and NetNews discussion groups using text-based interfaces to UNIX systems. Although many academic and administrative departments maintained Web sites, these exhibited little overall inte-

gration and were primarily organized around the school's administrative structure. The students wanted something easier to use that presented information in one place, arranged according to their needs.

While the students and staff working on the project explored several proprietary solutions, Wharton's IT staff believed that the Web and Internet technologies would provide a better long-term solution. But—back in 1994—the Web was fairly primitive. Web clients supported only basic HTML and server-side CGI calls. The first generation of Web browsers did not include Java, JavaScript, or Flash support. By enhancing the basic Web architecture with selected client tools and a custom-developed interface program, Wharton believed it could provide students with the functionality they sought while adhering to the “open” standards and scalable enterprise solutions of the emerging Internet technologies.

For the launch of the first version of SPIKE in 1995, Wharton developed Web-based interfaces to all of the school's key student services, switched its host-based e-mail to an IMAP-compliant system, and developed an integrated “launch pad” interface written in Visual Basic to tie it all together. (See Figure 1.) SPIKE quickly became the focal point for the school's communications with students.

SPIKE 2, launched in fall 1996, added features based on feedback from students and administrators, including the ability for students to customize the SPIKE interface. This feature was a result of what became known as the “ninth button problem.” A number of departments felt the eight primary “launch pad” buttons needed just *one more* button—that linked to their particular department or service. To address this requirement, the SPIKE interface expanded to include a secondary set of “launch pad” buttons, which could be customized to include links to additional online services of the student's choosing.

SPIKE 3 in fall 1997 brought significant changes to SPIKE's architecture and interface.² The Visual Basic program used for the SPIKE interface had become cumbersome. It was available only on the Windows platform, and the locally installed client program could not easily be updated after the software was distributed.

Fortunately, Web browsers had evolved significantly and by then included the ability to program the Web client using JavaScript. JavaScript and client-side cookies allowed the features and customization options of the previous versions to be expanded without a locally-installed client application. SPIKE 3 eliminated the Visual Basic program and replaced it with a purely browser-based interface. SPIKE's look

Figure 1**Original SPIKE 1 Interface**

also changed significantly, using much of the interface to present frequently updated content including a student calendar and a “What’s New” announcement service. (See Figure 2.)

SPIKE 4 included personalizable configurations for both MBA and undergraduate students and expanded the information broadcasting features beyond the Web with a collection of services Wharton dubbed “zero-click SPIKE”—SPIKE content delivered automatically to students. This was the era of “push” technology, and content was sent to screensavers and large-screen displays around campus. SPIKE 4 included both PointCast SmartScreens (using PointCast’s Intranet Broadcast Server) and Microsoft Active Channel screen-savers. These distribution options were updated in sync with the Web-based interface by a custom-developed tool known as the SPIKE Broadcast Server.

SPIKE 5 enhanced a number of SPIKE’s services and extended the information services beyond Wharton-specific content to include other information of interest to students, including the local weather forecast. A “fun stuff” toolbar, among other diversions, displayed randomly selected quotes from Wharton

students and faculty collected over the years by the *Wharton Journal*, the school’s student newspaper.

SPIKE 6 (2000) offered a slimmed

down version for Palm handheld devices, including custom AvantGo channels. It also gave students the ability to move SPIKE content into their personal information management tools, such as adding events from the calendar to their personal Microsoft Outlook calendar, Palm Datebook, or Yahoo Calendar account. Microsoft Outlook users had the option to replace the standard “Outlook Today” screen with a custom SPIKE Dashboard interface, which displayed personal information (such as the student’s individual calendar and e-mail inbox) alongside SPIKE’s student events calendar and “What’s New” announcements.

The seventh incarnation of SPIKE marked the most dramatic architectural overhaul since SPIKE 3. Using Adobe (then Macromedia) ColdFusion and Microsoft SQL server, MySPIKE, launched in 2001, was the most customizable version to date. (See Figure 3.)

A full-fledged information portal, MySPIKE included individualized information—such as each student’s class schedule—along with more general-purpose information such as weather and

Figure 2**Spike 3 with Web-Browser Interface**

Figure 3
MySPIKE in 2001



stock-market data. Students could rearrange the modules in the SPIKE interface and choose from a broad selection of general-information news feeds to further personalize the portal. Following versions added more features, including MyCareer, which added a personalized interface to Wharton's career management services.

The tenth version of SPIKE, SPIKE X, incorporated additional services that had evolved in parallel with SPIKE over the years, such as the Wharton Video Network's digital video content from Wharton courses. Services available in Wharton's new \$140 million learning facility, Jon M. Huntsman Hall, were also incorporated into the interface, such as the status of (and the ability to schedule) the building's 57 group study rooms.

Adobe (Macromedia) Flash components include a poll for instant feedback

on school issues. SPIKE Mobile automatically switches to a handheld-compatible version of the interface for users who access SPIKE from Palm, PocketPC, RIM Blackberry, or cell phone devices. RSS feeds of SPIKE's information services provide the option to add SPIKE content to blog aggregators and other personal information resources.

The current version, SPIKE 11, includes an SMS text-messaging interface that lets students view their course schedules or reserve group study rooms by sending a text message from their cell phones to a five-digit "short code" used for SPIKE services. SPIKE 11 also includes integrated message boards that automatically display recent messages and announcements in the SPIKE interface and an enhanced data analysis system for the school's Course Auction system. (See Figure 4.)

As of this writing, Wharton's SPIKE team is working with current Wharton students to develop the 12th annual version of SPIKE. SPIKE 12 will launch in fall 2006.

Lessons Learned

What lessons has Wharton gleaned from more than a decade of portal development? While there is no simple formula for building a successful Web portal, a few general principles emerged during the past 11 years of Wharton's intranet development.

■ *Reflect the world view of your audience.* The first, and perhaps most obvious, key to developing a successful intranet is to work collaboratively with your audience—faculty, administrators, and, most significantly, students—to reflect their world view. As obvious as this may sound, it is often overlooked. When the effort to develop an enterprise-wide portal comes from the enterprise itself, the final product often reflects that viewpoint. Forget about your organizational hierarchy and think about how your students look for information. Organize your intranet around their goals.

Also be wary of organizing your intranet around the underlying technological infrastructure. Developers have a tendency to group services by their underlying technologies—putting Web bulletin boards here, Usenet newsgroups there, and blogs over there. It's usually better to group these services by topic ("Course Discussions," "For Sale," and so on) rather than segregating them by technology.

In past years, Wharton students have even used Adobe Photoshop to mock up screenshots showing their vision of the next SPIKE interface.

■ *The customer is always quasi-right (do what they mean, not what they say).*

While you need to work closely with your target audience, there is a difference between listening to their goals and blindly following their mandates. In many cases, a request for new features comes not as a description of functionality but as a demand for

Figure 4

SPIKE 11 in 2005



a specific implementation. You'll hear statements like "We need to use technology *x* here." But perhaps technology *x* doesn't fit into your infrastructure, or maybe it's inappropriate for any number of reasons.

While you could explain all the reasons why technology *x* is a really bad choice, a better approach is to probe for the students' underlying needs. Ask "What capabilities would technology *x* give you?" You might find a better way of addressing their requirements.

■ *Balance the demands of your audience with those of your organization.*

While you should always start with the needs of your target audience, temper these with your judgment about the goals of the institution. Part of your job as a portal developer is to balance the requirements of different constituencies to construct a system that works for the enterprise as a whole.

With MySPIKE, for example, students could build their own interface by selecting which modules to display and their position on the screen. But two key modules—the student events calendar and the "What's New" announcements—could not be removed from the SPIKE interface. According to Bob Zarazowski, who led the team that developed MySPIKE,

School offices and student groups depended on [using] these as "broadcast" channels to reach all students. If [content] publishers weren't confident that an announcement to all MBA students would be seen by all MBA students, they would want to send the same announcement out as e-mail.

And the information clutter would only increase. Balancing what the audience wants with what will work at your institution is a key responsibility of the development group.

■ *Select a technology that lets you move quickly.*

Although you want to avoid organizing your intranet around the technology, that's not to say the technology doesn't matter. Selecting the right technology is key to the development process. Working collaboratively with your audience often means using "rapid development" techniques.

"Projects like this often don't have detailed specifications at the outset," explained Zarazowski. Tools that allow developers to quickly create prototypes that can then be modified or refined are essential.

According to Zarazowski, "ColdFusion lets us rapidly prototype and deploy each new version of SPIKE." Encapsulating routine functions (such as user authentication) into custom tags that developers can easily share permits quick development of new services. In addition, ColdFusion's tag-based markup paradigm lets both programmers and graphic designers modify code at any stage. While this "test it and see" style of development may be counter to some traditional structured programming practices, it often reflects real-world development cycles.

■ *Follow your audience: It's not just about the Web anymore.*

While we typically think of "portal" and "Web portal" as synonymous, that's outdated '90s thinking. Notice where your students spend their time—such as using their cell phones and PDAs. Follow your audience. Put your information where they're already looking rather than attempting to train them to look elsewhere.

"A lot of our students—particularly the incoming undergrads—are as comfortable using their cell phones for text messaging as they are for making phone calls," pointed out Dan Alig, who leads the current SPIKE development team.

■ *Keep content consistent across platforms.*

As you move your information onto multiple platforms, avoid fragmenting your content. The point is not simply to publish to handheld devices or large

plasma screens. You need to manage the information so the same content flows simultaneously to all these devices (albeit with a different presentation based on the characteristics of each medium).

Wharton CIO and Associate Dean Deirdre Woods noted that SPIKE is “more than just a Web site; [it’s] now an essential information management tool for both students and administrators at Wharton.”

As part of Wharton’s SPIKE project, the school developed a simple content publishing system back in the SPIKE 3 era (during the 1996–97 school year). Although the SPIKE Broadcast Server has gone through a number of architectural changes, the goal has been the same—to collect core content and distribute it simultaneously to various devices and presentation platforms.

■ *Centralize management, but decentralize input.*

While content synchronization may bring to mind a large, monolithic data management system, having a canonical source for each type of content doesn’t require a single, centralized system.

RSS (Really Simple Syndication) feeds and similar techniques make it easy to keep content in sync in a distributed environment. While selected core content (such as your school’s student calendar) may be maintained in a central database, other content could be generated by student blogs or independently managed discussion boards.

In fact, it’s important to keep the maintenance of the information close to the source of the content. Distributed departmental units and student organizations are the best sources of information about their groups.

There is, of course, a peril in this type of distributed content management if the presentation of the information becomes fragmented.

In the first few years of the SPIKE project, it was difficult to encourage content owners around the school to update SPIKE’s information services—departments and student groups were more focused on maintaining their own Web

sites than supporting a centralized publishing activity. Content syndication proved to be the key.

Once Wharton developers made custom views of SPIKE’s key information services—the student calendar and announcements services—easily syndicable on other Web sites, they had the best of both worlds. A department could, for example, use the SPIKE publishing tools to add an announcement to SPIKE that would be instantly syndicated to their departmental Web site. This ability to update their own sites provided the incentive for administrators and students around the school to use the publishing tools that also keep the portal current.

■ *Give it some pizzazz.*

A portal isn’t just another Web page. If successful, it will be a part of your organizational culture. Your portal needs to have an identity of its own.

Don’t be afraid to be a bit quirky. You’re talking to an internal audience—let the site reflect that fact. Giving the site an identity helps it stand out from the more formal Web content in your organization. Various versions of SPIKE have included humorous quotes from Wharton students and faculty, for example. And then there’s the SPIKE name³ and logo, which provide an identity for the varied components of the environment.

The details don’t matter. What’s important is to give your intranet service a unique identity that sets it apart from the content it aggregates and lets it enter the mainstream of your organizational culture.

■ *It ain’t over when it’s over (it’s all about the process).*

It’s tempting to focus your efforts on the technology that makes your portal possible or the features that make it functional. But it’s the process that drives its success. Building a portal is not the end point of the development cycle. It’s the beginning of the next phase of development.

Educational institutions have an advantage in that a significant percentage of their audience changes each year.

Incoming Wharton students don’t see their version of SPIKE as the end of years of refinement and enhancement—to them it’s just the foundation. As project leader Alig put it, “Typical feedback takes the form of ‘SPIKE is great. Now if it only did one more thing. . .’” And the development cycle begins for next year’s version.

The key lesson from more than a decade of intranet development is to view your enterprise portal not as a technology product but as an opportunity for ongoing engagement with your audience. As CIO Woods stated, “SPIKE is as much about process as it is technology. SPIKE is successful because it is constantly evolving to address the needs of Wharton’s students.” *e*

Endnotes

1. In *The Running of the Bulls* (New York: Gotham Books, 2005, p. 97), Nicole Ridgway noted, “A group of Wharton MBAs created the first business school intranet called SPIKE, which allowed for access to the Internet, messaging, and online student services, and would later add collaborative learning tools.”
2. For a contemporary (January 1998) description of the design goals of SPIKE 3, see Netscape’s *Intranet Executive* column “Wharton’s SPIKE: Using JavaScript to Develop an Enterprise Application” by Kendall Whitehouse, <<http://wp.netscape.com/columns/intranet/wharton.html>> (accessed November 8, 2005).
3. So why does Wharton call its intranet SPIKE? In an early planning meeting in 1994, SPIKE was chosen as a temporary code name for the project until a better name could be agreed upon. By the time the initial project launched, people liked the name SPIKE and decided to keep it. Wharton developed a logo based on a stock-market “fever graph.” After the fact the SPIKE team invented an acronym to explain the name: “Students’ Personalized Integrated Knowledge Environment,” but the acronym was soon dropped. SPIKE and the SPIKE logo are now registered trademarks of the university.

Kendall Whitehouse (whitehouse@wharton.upenn.edu) is Senior Director, Advanced Technology Development, at The Wharton School of the University of Pennsylvania in Philadelphia.