The Obama Campaign A Programmer's Perspective

WHAT WAS THE KEY TO THE OBAMA CAMPAIGN'S EFFECTIVENESS IN USING INFORMATION TECHNOLOGY?

Benjamin Boer, Consultant

On January 3, 2008, I sat in the boiler room waiting for the caucus to commence. At 7 p.m. the doors had been open for about an hour: months of preparation were coming to fruition. The phone calls had been made, volunteers had been canvassing, and now the moment had come. Could Barack Obama win the Iowa caucus?

The Obama Campaign

A Programmer's Perspective

Doors closed and the first text message came from a precinct: it looked like a large attendance. Then came the second, the third, the fourth. Each was typed into our model, and a projection was starting to form. The fifth, the sixth, and now the seventh. The projection of attendance seemed solid. The *Des Moines Register's* poll had been correct: more than 240,000 people participated in the Iowa caucus. The numbers were astounding, and Barack Obama was going to win it. Simple technologies—text messaging combined with data analysis and PHP—enabled the campaign to create a simple, yet effective, predictive dashboard.

The Obama campaign has been praised—with good reason—for its incredible use of technology. Many organizations would love to replicate its ability to do outreach, its focus on data, and its ability both to coordinate the efforts of hundreds of thousands of volunteers in a single direction and to empower those individuals to take control of their own distinct parts of the campaign.

The use of technology within the Obama campaign creates two seemingly contradictory points: the technology strategy was not a technology strategy—it was an overall strategy—yet it could not have been executed without technology. But this misses what programmers have always understood about software—a truth that has finally blossomed in the age of social networking: software itself is an organizing force that equips organizations to achieve their goals. The Obama campaign used technology as a front-end enabler rather than a back-end support, and this synchronization between mission and tools allowed for the amplification of both.

While the Internet operations of the campaign represented the most visible aspect of its technology efforts, the campaign used technology throughout its operations: for internal processes, for opening offices, for managing GOTV (get-out-the-vote) operations, and for communicating with internal and external stakeholders, including fund raisers, policy advisors, and external consultants. Much of this technology was integrated so that fund raising could be linked across different channels (offline and

online) and GOTV efforts could be executed using both online resources and old-fashioned canvassing. All of this had to be linked to the compliance and operation efforts of the campaign while making sure that the budget was managed and every dollar accounted for. All of these systems had to be created, integrated, and rolled out with training for thousands of staff and volunteers in less than two years.

NO MAGIC FORMULA

How was all this technology designed, developed, and rolled out in such a short time (and at a cost that reflected its short-term nature)? To understand the success of the Obama campaign in using technology, it is important to reflect on how other software architectures have empowered and reflected the organizations they have supported. For example, command-and-control organizations used the structured programming architectures of their time. Object-oriented architectures reflected the move to shared-services organizations in which functions such as HR and finance were moved to more central parts of the organization. Service-oriented architectures seem to support the outsourcing that many organizations have moved toward, where access to the objects is necessary both outside and inside the organizations. Successful organizations have been able to integrate the most current software architecture patterns into their business model quickly and effectively. At this juncture, where technology and people are moving in the same direction, innovation takes place.

The Obama campaign reflects this ability to its extreme. What have been the advances in software architecture that made the Obama campaign successful? Obviously, social networking played a huge part in organizing people, but other models that are important to look at are the open source development and PaaS and SaaS (Platform and Software as a Service) models. Additionally, the campaign made extensive use of data analysis. Combining these pieces through the use of existing tools allowed the campaign to innovate. Of course, these concepts had constraints, such as a high learning curve, the need for a

significant workforce, or a limited user base. But because the *people* were in sync with the concept of grassroots experimentation, when a concept was successful, it was nurtured and the resources were provided to expand it.

How did the architectural elements of the technology reflect the overall principles of the organization and campaign? Open source development, with its focus on distributing the ability of developers to add to an existing code base in a controlled but expansive manner, reflected the campaign's dependence on a far-flung set of leaders and volunteers. People in both universes brought their unique talents to the project. Equally as important, when a set of tasks was not accomplished as planned, resources could be moved to the problem at hand. In the same way that volunteers flowed from making calls in Pennsylvania to making them in Indiana, developers were able to flow from data exchange to the call tool.

An example of this flexibility was the data program for certain primary dates. No one knew up front exactly how the campaign was going to play out. Not all the primary dates were given the same emphasis, but as certain days approached, the need for data programs and GOTV efforts steadily became more apparent. Each state organization had been given great leeway in designing how it was going to use the available systems, and ad hoc development of scripts and extensions of the systems were necessary as different teams attempted to stretch the data systems. For each primary, however, the relevant team was reconfigured, bringing best practices in from many locations and refactoring and consolidating processes that had been developed for targeting and scoring voters. This willingness to experiment with data analysis and then expand its use is indicative of how the campaign operated.

Working in conjunction with these individual state-initiated designs were the existing platforms for development. Just as the campaign rallied around a core set of principles and was able to incorporate these into everything the campaign did, the use of multiple platforms for GOTV, the Web site, finance, operations, and communications allowed for experimentation within a controlled environment. Ideas could be tried, tested, and changed. Once an idea proved successful, it could be expanded and rolled out to thousands or hundreds of thousands of people with incredible speed.

PaaS technology made this type of experimentation straightforward. Platforms that could be easily configured allowed operations quickly to move processes, such as hiring and procurement, from headquarters to the hundreds of offices that were eventually established,

yet allowed for centralized control of these processes. In other instances, programs such as Precinct Captain were designed in the state offices using simple platform tools and then extended to other programs with more concerted development efforts.

Of course, there were typical technology issues, such as systems that got more traffic than expected and offices that did not have quite the right configuration. But the ability to be inventive and monitor results and understand the core elements of each of the implemented systems allowed the campaign to compensate for these issues. In addition, backup plans—often very simple ones—were in place. For example, having to depend on data systems that had not seen the scale of an election night was risky; consequently, much of the data was loaded into parallel secondary systems. Similarly, understanding which parts of the Web site were critical allowed the campaign to refocus its resources when necessary. Preparation of appropriate communications and operating plans was key to the flexibility that allowed people who understood the systems best to make decisions and trust that these decisions would be made in the best interests of the campaign.

Innovation is a buzzword that is used too often, and organizations too often are looking to understand the “magic” formula. Still, in the case of the Obama campaign, few would deny that it was the most innovative organization they have ever seen. The constraints that were forced upon the campaign at its inception may have been integral to making this happen. Knowing that the campaign would exist for only two years and (initially) that it was the underdog and not as well funded as its competitors led to an inventiveness and willingness to experiment and a distribution of power that might not otherwise have been possible. Ultimately, existing technologies that allowed for distributed development were combined with an adherence to a core set of architectures and principles. This meant that ideas could be quickly and effectively rolled out to the many people who needed them. ◻

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

BEN BOER is the former VP of technology for AHA! interactive, an educational software company. He currently provides technology and policy consulting for nonprofits and was a technology consultant to Obama for America.

© 2009 ACM 1542-7730/09/0100 \$5.00