NINETEEN

Archive Experiences: A Vision for User-Centered Design in the Digital Humanities

LIZA POTTS

Over the past several decades, we have witnessed a race to build, archive, and distribute various scholarly materials across the digital humanities (Earhart 2012). While these systems hold a veritable treasure trove of knowledge, they are crippled by their user experiences. Instead of distributing knowledge to the public and encouraging scholarly exploration, interactions with these systems are clunky at best and irrelevant at worse. Rather than building systems that prioritize data above experience, we need to architect archives that are focused on engagement with scholars and outreach with the public. This is a call for scholar practitioners in rhetoric to engage with the digital humanities as user advocates, experience architects, and participantcentered researchers.

Many of these digital humanities systems, either by accident or by design, are focused on serving up material—images, texts, and videos—rather than engaging with participants. What these archives in practice and the digital humanities in general desperately need is a sense of audience, appeal, and interaction. Instead, these archives are often inwardly facing, aimed at their own research partners and, perhaps, their own specific field. However, these systems are live and on the public Internet, where there is a huge opportunity to interact with and create new audiences. Unfortunately, these systems often lack user-centered, rhetorically situated, contextually aware experiences. We need to build systems that are more widely accessible.

To make such a move toward accessibility, we need to build digital humanities projects that allow for more engagement with at a minimum the digital humanities specifically and a multitude of audiences more broadly. We must design and build for experience. I refer to these practices, rooted firmly in technical communication and evolving from user-centered design (UCD), as *experience architecture*. Experience architecture is an emerging practice, one that draws together issues of information design, information architecture, interaction design, and usability studies to assess and build products, services, and processes. The outcomes of a well-architected system include systems, interfaces, and policies that support participation, growth, and sustainability—in other words, building experiences that are focused on human experience, the kinds of experiences we are espousing when we discuss the application of rhetoric and digital humanities. By realigning project teams across disciplines to build user-centered experiences, we can have a huge impact on how these systems are received by their audiences.

Digital Humanities and the Hidden Users/Participants

The need for refactoring, rebuilding, and integrating experience architecture into the product life cycle of these projects is clear. When scholars in digital humanities turn to examining usability and design, it is often to comment on design rather than engage with its practice. Examining the most recent major book surveying the digital humanities (Gold 2012) shows that only two of the forty-nine chapters and printed blog posts relate even somewhat to UCD and that both are in the critique section of the text (Edwards 2012; Williams 2012). Similarly, recent publications coming out of a major digital visualization lab makes no mention of the terms *usability* or *UCD*, much less any of the necessary concepts behind building user-centered, rather than data-driven, experiences (Manovich 2012, n.d.). Searching one of the largest digital humanities hubs shows that only a minority of scholar practitioners self-identify as designers, with no category available for expertise in information architecture or usability (Digital Humanities Commons, n.d.). That said, in rhetoric there is a much stronger understanding of and skill set for UCD than is being represented in these publications. This work is

already taking place. What is needed are more obvious connections between our work in technical communication and work across other fields in digital humanities. Our years of experience in UCD, information design, and experience architecture are an asset for digital humanities projects.

Collections on digital humanities scholarship contain some brief mentions of users and design. Some even suggested readings that are woefully dated, continuing to mention early works on UCD from, e.g., Norman (1989), Cooper (1995), and Nielsen (1993). We are now on the cusp of a new age, where big data, metadata, and participatory culture are set to collide. And while Norman, Cooper, and friends were certainly useful companions to help give us a foundation, we now need to pay attention to new ways of architecting, designing, managing, and developing our products. There is a wave of new work all espousing a collective discussion on how to engage users as participants, both in design and in social engagement, that I will discuss in the next section. Suffice it to say, we have moved forward in industry practice and research in our field.

Another major issue is how projects are managed. While industry practice has moved into agile methods and practices, many digital humanities projects seem to be trapped in waterfall methods of software development. Some of these projects even seem to promote this dated process. Many of the discussions of making fetishize the concept of coding, which is disconcerting for several reasons—not the least because it feels like a repeat of a era we have already lived through. These movements are, indeed, akin to the late 1990s of computing, making the above-mentioned references of dated UCD work relevant—albeit entirely out of place given current workplace practices. Rather than relive the dot-com boom and bust, we need to call for an intervention, redirecting and updating the practices in digital humanities projects. In the next section, I will define the kinds of work involved in UCD, calling on us to be the experience architects that are so woefully missing on many of these digital humanities projects.

Rather than seeing these gaps as a lack of interest, we can see this moment as an opportunity for our field. And, frankly, it is unfair to expect our colleagues in English, history, linguistics, and other humanities-focused fields to have the same backgrounds we have in information design, usability, information architecture, and other technical communication teachings. Realistically, we are the ones who have the backgrounds, teachings, and experiences to lead projects. Scholar practitioners in rhetoric are well equipped to respond to this challenge because of their deep tradition of user advocacy and empowerment design (Davis 2002; Hart-Davidson 2001; Grabill and Simmons 1998; Johnson 1998; Mirel 1996; Miller 1979). This moment in this field is one in which a shift toward participating in and researching the design of interfaces, interactions, and processes and analyzing the use of these genres and associated tools is becoming more prevalent (Spinuzzi 2003, 2007, 2013; Potts and Jones 2011; Swarts 2008). Moreover, all these developments contribute to our abilities to work effectively in complex contexts where one must address these issues of sustainability, participation, and engagement. This lack of UCD in digital humanities projects provides an opportunity for us to fill this gap, participate in these programs, and draw on our strengths to engage with these scholars and materials to refocus these projects in ways that support UCD.

Rhetoric and UCD

In making these connections between rhetoric and the digital humanities, we can focus on engagement. By *engagement*, I mean engagement both with the internal digital humanities communities as they exist today and with the external audiences for digital humanities projects. Concentrating on building systems that provide for engagement between the curators and the participants can broaden the audiences for these projects, addressing the needs of both researchers and the public. Participating in these projects as experience architects, usability experts, information architects, and information designers, we can propose early questions about audience, purpose, and scope.

These tasks are the same kind of work that we are already doing in our own practices and classrooms. Our worldview is unique to those of us trained in rhetoric in general and certainly to those of us with experience running technical projects and mentoring our students through these activities. Theory and practices for UCD are not widely known throughout the humanities. It is an opportunity and a responsibility to lead and participate in these projects because of our knowledge of how to architect, manage, and improve both the process and the building of these products and services.

We may need to clearly make the case for UCD projects, distributing the knowledge that we already have in our field to those outside it. We may find ourselves having the same arguments as we did in the 1990s (why do we need UCD?). Thankfully, we have a rich set of materials from which we can draw on old and new discussions about user interface design (Tidwell 2011), information architecture (Resmini and Rosati 2011; Morville and Rosenfeld 2006), content strategy (Halvorson and Rach 2012; Redish 2007), agile development (Ratcliffe and McNeill 2012), project management (Berkun 2008), and team management (Lund 2011).

Engaging with users and participants, we can work on a strategy for the success of digital humanities projects. There are many excellent examples out there for guiding principles of user experience, including Morville's (2005) concept of findability, Krug's (2006) thoughts on not thinking, and Garret's (2010) levels of interaction and work flow. All three books are good for understanding the basic concepts of UCD. As another example, we can look to Halvorson and Rach's (2012) work on experience and content, where they outline three major foci: core, content, and people. Purpose, context, and audience are major concepts in rhetoric, ones that we emphasize repeatedly within our pedagogy and our practice. Looking at a combination of these components, we can discuss what UCD can and should be for digital humanities, focused squarely on creating contextualized experiences for engagement. Launching any new project—whether it is creating a new product, service, process, or whatever—requires asking questions up front to ensure successful outcomes. And, while we ask these questions in the beginning, it is important to revisit them constantly to align the project with the goals of our users and stakeholders.

Purpose

Before diving into any project, we are obligated to ask what we are trying to achieve. What is the purpose of this digital archive, this mobile application, this Web site, this kiosk, etc.? Coming to some consensus on what the end result is meant to achieve is of critical importance. There are some key purposes to digital humanities projects: to share knowledge, to educate the public, to appeal to donors, to connect to a wider research community, etc. While any project can have multiple purposes, it is important to come to an agreement on its major purpose. Aligning the team toward this goal is essential to building a product that will be user centered.

Content

There are many types of content on digital humanities Web sites and applications, including images, games, videos, text, pdfs, data sets, maps, and audio. The types of content and the ways in which it is presented all need to take into consideration the context in which our users/participants will engage with it. Many digital humanities projects are addressed and logically present their work to their peers and funders. Credibility is heavily emphasized, as evidenced by the listings of the researchers, contributors, centers, and funding agencies that contributed to the project. That said, the connection between producer and potential audience is not always as overt. If this content is of public interest, and most digital humanities research certainly can be, then how can we best present this content to these audiences? What entry points will we create to these rich materials? How can we create the kinds of appeals that connect site producers with participants (academic and the general public) who are eager to engage with this content? Building content and structures to serve that content in interactive, contextualized ways will help distribute our research and create experiences for multiple audiences.

People

Whenever we sit down to start a project, the first question I have is, Who is our audience? Understanding audience is a key skill to develop in our students. The next question is always, What is the context in which that audience will use our product/service/process? For digital humanities projects, there is a clear and strong audience of scholars. And, while that is important, there is a set of audiences out there that are interested in our work, can engage with our work, and can add value to our research. This group is composed, at a bare minimum, of the general public, prospective benefactors, and interdisciplinary researchers. So why is it that so many of these archives and projects lack any sense of

audience, interaction, and appeal? By understanding our audiences, we can create interactive, useful, and rich experiences for them.

Having a grasp on these three core concepts (purpose, content, people) is essential for developing products focused on users and participants. There are many methods for developing user-centered products, services, and processes. Useful methods include landscape analysis, content inventory, focus groups, site visits, contextual inquiry, surveys, researcher participation, affinity diagramming, card sorting, prototyping, and usability testing. Stakeholder interviews can help us gain a better understanding of the problem space as well as of the goals of the project. Sitting down and discussing the reasons for the project, the hopes for new solutions, and the constraints in which the project will be deployed help focus the project. Contextual inquiry allows us to visit the spaces in which these technologies are (or will be) deployed. Observing task flows, learning about usage, and witnessing patterns of behavior can help document work flow and lead to solution discovery. Returning back to the team armed with these data, we can begin to design useful prototypes. After the prototype is built, we can conduct usability testing. The results of these tests can help us refine the final implementation as well as create wish lists for future iterations based on user feedback.

This list is by no means comprehensive, and there are countless Web sites, articles, and books on further methods that could help with our research work. I have listed the ones that have served me best and will, I think, give us a good start, but leaders will need to select the methods that work for specific projects and teams. These are iterative processes—meaning that we can design, share the design with the internal team, iterate on the design, show the users/participants the design, iterate some more, etc. Above all else, it is important to work on architecting and designing these systems before incurring the high costs of development work. To do otherwise is costly and time consuming and will certainly result in user-experience issues.

Future Work as Agents of Social Change

We can bring a vision of UCD to the digital humanities that stands on the shoulders of rhetoric's vast research on communication design, information architecture, and rhetoric. Recently, an industry contact asked me for an updated reading list and set of examples for UCD. He stated plainly that industry looks to universities to be on the cutting edge of technologies, of movements, of change. It is imperative that the products and services coming out of the digital humanities are cuttingedge. With a background in rhetoric and training in UCD, we can be the "agent of social change" (Savage 2004, 183). We must move on this moment and architect for experience, rather than simply archiving collections. Such moves will allow us to explore what Kathleen Fitzpatrick refers to as the creative tension between "making and interpreting, between the field's history and its future" (2012, 14). It is within this space that we may also see the future of digital humanities and rhetoric, one in which we can be key contributors, leading the work to "shape emerging digital culture rather than only adapt to the change it brings" (Salvo and Rosinski 2010, 111).

References

Berkun, Scott. 2008. *Making Things Happen: Mastering Project Management*. Sebastopol: O'Reilly.

Cooper, Alan. 1995. *About Face: The Essentials of User Interface Design*. Foster City, CA: IDG Worldwide.

Davis, Marjorie T. 2002. "Shaping the Future of Our Profession." *Technical Communication* 48:139–44.

Digital Humanities Commons. n.d. "Collaborators." http://dhcommons.org/collaborators.

Earhart, Amy E. 2012. "Can Information Be Unfettered? Race and the New Digital Humanities Canon." In *Debates in the Digital Humanities*, ed. Matthew K. Gold, 309–18. Minneapolis: University of Minnesota Press.

Edwards, Charlie. 2012. "The Digital Humanities and Its Users." In *Debates in the Digital Humanities*, ed. Matthew K. Gold, 213–32. Minneapolis: University of Minnesota Press.

Fitzpatrick, Kathleen. 2012. "The Humanities, Done Digitally." In *Debates in the Digital Humanities*, ed. Matthew K. Gold, 12–15. Minneapolis: University of Minnesota Press.

Garrett, Jesse James. 2010. *The Elements of User Experience: User-Centered Design for the Web*. 2nd ed. Indianapolis: New Riders.

Gold, Matthew K., ed. 2012. *Debates in the Digital Humanities*. Minneapolis: University of Minnesota Press.

Grabill, Jeffrey T., and W. Michele Simmons. 1998. "Toward a Critical Rhetoric of Risk Communication: Producing Citizens and the Role of Technical Communication." *Technical Communication Quarterly* 7:415–41.

Halvorson, Kristina, and Melissa Rach. 2012. *Content Strategy for the Web*. 2nd ed. Berkeley, CA: New Riders.

Hart-Davidson, William. 2001. “On Writing, Technical Communication, and Information Technology: The Core Competencies of Technical Communication.” *Technical Communication* 48:145–55.
Johnson, Robert R. 1998. *User-Centered Technology: A Rhetorical Theory for Computers and Other Mundane Artifacts*. Albany: State University of New York Press.
Krug, Steve. 2006. *Don’t Make Me Think! A Common Sense Approach to Web Usability*. 2nd ed. Berkeley, CA: New Riders.
Lund, Arnie. 2011. *User Experience Management: Essential Skills for Leading Effective UX Teams*. Burlington, MA: Morgan Kaufmann.
Manovich, Lev. 2012. “Trending: The Promises and Challenges of Big Social Data.” In *Debates in the Digital Humanities* , ed. Matthew K. Gold, 460–73. Minneapolis: University of Minnesota Press.
———. n.d. “Visualizing Image and Video Collections: Examples.” *Media Visualization*. http://softwarestudies.com /cultural_analytics/Visualizing-image-and-video-collections-examples.pdf.
Miller, Carolyn R. 1979. “A Humanistic Rationale for Technical Writing.” *College English* 40:610–17.
Mirel, Barbara. 1996. “Writing and Database Technology: Extending the Definition of Writing in the Workplace.” In *Electronic Literacies in the Workplace: Technologies of Writing*, ed. Patricia Sullivan and Jennie Dautermann, 91–114. Urbana: NCTE/Computers and Composition.
Morville, Peter. 2005. *Ambient Findability: What We Find Changes Who We Become*. Sebastopol, CA: O’Reilly.
Morville, Peter, and Louis Rosenfeld. 2006. *Information Architecture for the World Wide Web: Designing Large-Scale Web Sites*. 3rd e. Sebastopol: O’Reilly.
Nielsen, Jakob. 1993. *Usability Engineering*. San Diego: Morgan Kaufmann.
Norman, Donald A. 1989. *The Design of Everyday Things*. New York: Doubleday.
Potts, Liza, and Dave Jones. 2011. “Contextualizing Experiences: Tracing the Relationships between People and Technologies in the Social Web.” *Journal of Business and Technical Communication* 25:1–21.
Ratcliffe, Lindsay, and Marc McNeill. 2012. *Agile Experience Design: A Digital Designer’s Guide to Agile, Lean, and Continuous*. Berkeley, CA: New Riders.
Redish, Janice. 2007. *Letting Go of the Words: Writing Web Content that Works*. Amsterdam: Morgan Kaufmann.
Resmini, Andrea, and Luca Rosati. 2011. *Pervasive Information Architecture: Designing Cross-Channel User Experiences*. Burlington, MA: Morgan Kaufmann.
Salvo, Michael J., and Paula Rosinski. 2010. “Information Design: From Authoring Text to Architecting Virtual Space.” In *Digital Literacy for Technical Communication*, ed. Rachel Spilka, 103–27. New York: Routledge.
Savage, Gerald J. 2004. “Tricksters, Fools, and Sophists: Technical Communication as Postmodern Rhetoric.” In *Power and Legitimacy in Technical Communication*, vol. 2, *Strategies for Professional Status*, ed. Teresa Kynell-Hunt and Gerald J. Savage, 167–93. Amityville, NY: Baywood.
Spinuzzi, Clay. 2003. *Tracing Genres through Organizations: A Sociocultural Approach to Information Design*. Cambridge, MA: MIT Press.
———. 2007. “Guest Editor’s Introduction: Technical Communication in the Age of Distributed Work.” *Technical Communication Quarterly* 16:265–77.
———. 2013. *Topsight: A Guide to Studying, Diagnosing, and Fixing Information Flow in Organizations*. N.p.: CreateSpace Independent Publishing Platform.
Swarts, Jason. 2008. *Together with Technology: Writing Review, Enculturation, and Technological Mediation*. Amityville, NY: Baywood.
Tidwell, Jenifer. 2011. *Designing Interfaces*. 2nd ed. Sebastopol: O’Reilly.
Williams, George H. 2012. “Disability, Universal Design, and the Digital.” In *Debates in the Digital Humanities*, ed. Matthew K. Gold, 202–12. Minneapolis: University of Minnesota Press.

TWENTY

MVC, Materiality, and the Magus: The Rhetoric of Source-Level Production

KARL STOLLEY

> The greater the range and intenseness of the opportunities for the exercising of our symbolic prowess, the greater might be our delight in such modes of action.
> **KENNETH BURKE, *LANGUAGE ASYMBOLIC ACTION***

Let me open with the major premise of this chapter's argument: programming is writing. I mean that literally, as I will illustrate with some limited examples from the Ruby on Rails framework in this chapter. Though I mean the phrase literally, *programming is writing* is frequently invoked as a metaphor, even among programmers. I agree with programmer Steve McConnell (2004), who dismisses writing as "the most primitive metaphor for software development" (7). However, there are some key points in his dismissal that are unsound. Among McConnell's observations on the metaphor's shortcomings:

- "[Writing] doesn't require any formal planning, and you figure out what you want to say as you go" (5).
- "Writing is usually a one-person activity" (6).
- "In writing, a high premium is placed on originality. In software construction, [originality] is often less effective than focusing on the reuse of design ideas, code, and test cases from previous artifacts" (6).

An oversimplified conception of writing forms the basis of McConnell's critique of the metaphor. True, writing may not require formal planning, but, as even the introductory-writing student quickly discovers, that is an unstudied and ineffective way to proceed. Playwrights, poets, and technical writers alike know that writing is collaborative, to some degree, always. And, from citation practices to genre features, writers have a wide foundation of reusable material on which originality is built, given a particular rhetorical occasion: to write is to engage in intelligent, ethical (i.e., nonplagiarized) reuse of previous artifacts. However, as I discuss later, richer encounters with originality that digital writers and humanists alike might otherwise experience are routinely preempted by careless reuse and outsourcing of source-level production.

To provide a glimpse into the experience of programming, I will present some of the activities involved in building a Web application using Ruby on Rails (a full Rails app is available with additional commentary via this book's companion Web site: www.press.uchicago.edu/sites/rdh/). There is not room in this chapter for a full technical description of Ruby on Rails, but it is enough for now to note that Rails is an open-source Web-application development framework written in the Ruby programming language.

Rails can be installed, invoked, and developed entirely through writing. There is no file to manually download and unzip (as with Drupal, WordPress, and other platforms as frameworks that are currently popular) and nothing to click on. Rails is installed by running a command on any system with a command-line interface (CLI; indicated here by the dollar sign, $) and a Ruby installation, as is the case for Mac OS X and many distributions of Linux:

```
$ gem install rails
```

Installing Rails does not create a Rails application. Among other things detailed below, a Rails installation includes a command-line program, conveniently called *rails*; and it is with the rails command that a project is brought into existence. All that Rails requires is a name for the project, and in this example I have chosen BeSocial, an imaginary social networking application:

```
$ rails new besocial
```