Developing a Digital Project (With Omeka)

Markus Wust
Brian Norberg
Grant Glass
Welcome to DHSI 2018!

Thanks for joining the DHSI community!

In this booklet, you will find essential course materials prefaced by some useful information about getting settled initially at UVic, finding your way around, getting logged in to our network (after you’ve registered the day before our courses begin), and so on.

Given our community’s focus on things computational, it will be a surprise to no one that we might expect additional information online for some of the classes - your instructors will let you know - or that the most current version of all DHSI-related information may be found on our website at dhsi.org.

To access the DHSI wifi network, simply go into your wireless settings and connect to the “DHSI” network and enter the password “dhsi2018”.

And please don’t hesitate to be in touch with us at institut@uvic.ca or via Twitter at @AlyssaA_DHSI or @DHInstitute if we can be of any help ....
Regional Map of Greater Victoria

Legend
- **Direction of Travel**
- **Route Name**
- **Transit Exchange**
- **Park & Ride Lot** (no overnight parking)
- **Major Stop**

**Average Frequency**

- **Regional Route**: 15–60 minute service with limited stops
- **Frequent Route**: 15 minute or better service, 7am-7pm, Mon-Fri
- **Local Route**: 20–120 minute service

**Legend**

- **UVic**
- **Regional Map of Greater Victoria**
- **Esquimalt Harbour**
- **Dockyard**
- **Vic General**
- **Colwood**
- **Esquimalt**
- **Esquimalt Plaza**
- **Ministry of Finance Building**
- **Royal Oak**
- **Royal Oak Shopping Centre**
- **UVic X**
- **Regional Route**: 20–120 minute service
- **Frequent Route**: 15 minute or better service, 7am-7pm, Mon-Fri
- **Local Route**: 15–60 minute service with limited stops
The 2018 schedule is just about ready! A very few things to confirm, add, etc, but this is the place to be to find out what is happening when / where ...

---

Sunday, 3 June 2018 [DHSI Registration + Suggested Outings]

If you're here a day or two before we begin, or staying a day or two afterwards, here are a few ideas of things you might consider doing ...

▼ Suggested Outing 1, Botanical Beach (self-organised; car needed)

A self-guided visit to the wet, wild west coast tidal shelf (and historically-significant former research site) at Botanical Beach: we recommend departing early (around 8.00 am) to catch low tide for a better view of the wonderful undersea life! Consider带来 a packed lunch to nibble-on while watching the crashing waves when there, and then have an afternoon drink enjoying the view from the deck of the Port Renfrew Hotel.

▼ Suggested Outing 2, Butchart Gardens (self-organised)

A shorter journey to the resplendently beautiful Butchart Gardens and, if you like, followed by (ahem) a few minutes at the nearby Church and State Winery, in the Saanich Peninsula. About an hour there by public bus from UVic, or 30 minutes by car.

▼ Suggested Outing 3, Saltspring Island (self-organised; a full day, car/bus + ferry combo)

Why not take a day to explore and celebrate the funky, laid back, Canadian gulf island lifestyle on Saltspring Island. Ferry departs regularly from the Schwartz Bay ferry terminal, which is about one hour by bus / 30 minutes by car from UVic. You may decide to stay on forever ....

▼ Suggested Outing 4, Paddling Victoria's Inner Harbour (self-organised)

A shorter time, seeing Victoria's beautiful city centre from the waterways that initially inspired its foundation. A great choice if the day is sunny and warm. Canoes, kayaks, and paddle boards are readily rented from Ocean River Adventures and conveniently launched from right behind the store. Very chill.

And more!

Self-organised High Tea at the Empress Hotel, scooter rentals, visit to the Royal BC Museum, darts at Christies Carriage House, a hangry breakfast at a local diner, whale watching, kayaking, brew pub sampling (at Spinnaker's, Swans, Moon Under Water, and beyond!), paddle-boarding, a tour of used bookstores, and more have also been suggested!

Psst: Some Suggested Outings

9:00 to 4:00

▼ Early Class Meeting: 4. [Foundations] DH For Department Chairs and Deans (Hickman 120, Classroom)

Further details are available from instructors in mid May to those registered in the class. Registration materials will be available in the classroom.

3:00 to 5:00

DHSI Registration (MacLaurin Building, Room A100)

After registration, many will wander to Cadboro Bay and the pub at Smuggler's Cove OR the other direction to Shelbourne Plaza and Maude Hunter's Pub OR even into the city for a nice meal.

Monday, 4 June 2018

Your hosts for the week are Alyssa Arbuckle, Ray Siemens, and Dan Sondheim.

7:45 to 8:15

Last-minute Registration (MacLaurin Building, Room A100)

8:30 to 10:00

Welcome, Orientation, and Instructor Overview (MacLaurin A144)
Classes in Session (click for details and locations)

3. [Foundations] Making Choices About Your Data (MacLaurin D109, Classroom)
4. [Foundations] DH For Department Chairs and Deans (Hickman 120, Classroom)
5. [Foundations] Introduction to Javascript and Data Visualization (Clearihue D132, Classroom)
6. [Foundations] Introduction to Computation for Literary Criticism (Clearihue A105, Lab)
7. Out-of-the-Box Text Analysis for the Digital Humanities (Human and Social Development A160, Lab)
8. Sounds and Digital Humanities (MacLaurin D111, Classroom)
9. Digital Humanities Pedagogy: Integration in the Curriculum (MacLaurin D016, Classroom)
10. Text Processing - Techniques & Traditions (McPherson Library A003, Classroom)
11. 3D Modelling for the Digital Humanities and Social Sciences (MacLaurin D010, Classroom)
12. Conceptualising and Creating a Digital Edition (MacLaurin D103, Classroom)
13. Visualizing Information: Where Data Meets Design (MacLaurin D107, Classroom)
14. Introduction to Electronic Literature in DH: Research and Practice (MacLaurin D115, Classroom)
15. Race, Social Justice, and DH: Applied Theories and Methods (MacLaurin D105, Classroom)
16. XML Applications for Historical and Literary Research (Clearihue A103, Lab)
17. Processing Humanities Multimedia (Human and Social Development A150, Lab)
18. Web APIs with Python (Human and Social Development A170, Lab)
19. Digital Games as Tools for Scholarly Research, Communication and Pedagogy (MacLaurin D110, Classroom)
20. Ethical Data Visualization: Taming Treacherous Data (MacLaurin D101, Classroom)
21. Digital Publishing in the Humanities (Clearihue D131, Classroom)
22. Linked Open Data and the Semantic Web (Clearihue D130, Classroom)
23. Introduction to IIIF: Sharing, Consuming, and Annotating the World’s Images (MacLaurin D114, Classroom)
24. Feminist Digital Humanities: Theoretical, Social, and Material Engagements (Cornett A229, Classroom)
25. The Frontend: Modern JavaScript & CSS Development (Clearihue A030, Classroom)

Lunch break / Unconference Coordination Session (MacLaurin A144)
(Grab a sandwich and come on down!)

Undergraduate Meet-up, Brown-Bag (details via email)

Classes in Session

Institute Panel: Perspectives on DH (or, #myDHis …)
Chair: Alyssa Arbuckle (U Victoria)
(MacLaurin A144)

Milena Radzikowska (Mt Royal C): "Release the Kraken: Story-Driven Prototyping for the Digital Humanities."
Abstract: I have spent the last 15 years of my career designing text analysis tools for use by humanities scholars. In this brief presentation, I propose to share a concept-based approach to interface design for DH.

Emily Murphy (U Victoria): "#MyDHis Edgy."
Abstract: I will build upon—or, possibly, perform a misprision of—a tweet by Polina Vinogradova; "#myDHis messy, dusty, edgy, and radically inclusive!" Vinogradova evokes the mess and dust of the archives, the edges that connect nodes of a network, and the political impetus to think of cultural history and community together. I argue that these aspects of DH have a renewed importance as we head into a moment of feminist historiography.

Margaret Konkol (Old Dominion U): "Prototyping Mina Loy’s Alphabet with a 3D Printer."
Abstract: This talk discusses the interpretive and methodological implications of using 3D printing technologies to prototype the archival diagrams of a proposed but never constructed plastic segmental alphabet letter kit—a game designed by modernist poet Mina Loy for F.A.O Schwarz. Although intended as a toy for young children, "The Alphabet that Builds Itself," as a work of "object typography" articulates a theory of language as kinetic, geometric, recombinant, and open to mutation. Alphabetic segments extend into the x, y, and z coordinates in exponential iterations and conjoin with magnets. Combining elements of contemporaneous typefaces like Futura and Gill Sans, which represented modernity’s functional ideals and democratic principles of simplicity, these recombinant letters represent, as this talk argues, Loy’s unpublished modernist poem, an articulation of Loy’s concept of language as a physical fact in which substance, not just form, is semantic.

Lee Zickel (Case Western Reserve U): "Comfortably Trepid."
Abstract: #myDHi found outside the well-established, DH-friendly institutions, at an institution that is devoted predominantly to Medicine and Engineering. I, with increasing frequency other DH practitioners and instructors, am not positioned in a DH Lab or Humanities Center, but in ITS. Part teacher, part technologist, part translator, I will briefly discuss my work supporting humanists and social scientists, particularly those who are new to or less comfortable with computational methodologies.

Dorothy Kim (Vassar C): "#MyDHis Antifascist."
Abstract: I've spent a lot of time in the last 12 months thinking about fascism, digital humanities, its long histories, and what it means to do DH work that centers social justice particularly in this global rise of late fascism. I will speak briefly about DH's history, including the medieval history related to Busa but how that history really connects to data systems that created the Holocaust and also participated in the Cold War nuclear military complex.
Randa El Khatib (U Victoria): "Learning from the Iterative Process."
Abstract: #MyDHis Iterative. In addition to the improvements that come with iterative projects, the iterative process itself is a fruitful area for scholarly inquiry. Within this iterative context, the various teams that I work with and I have been reflecting on and rethinking central DH practices, such as what it means to collaborate, prototype, remix, and implement DH values in our work. In this talk, I will present the various lessons learnt along the way.

Sarah Melton (Boston C): "#MyDHis...People."
Abstract: Taking seriously Miriam Posner’s exhortation to “commit to DH people, not DH projects,” I invite us to reflect on how people are the core of DH. In this brief talk, I will explore the intersections between DH, labor, and infrastructure.

5:00 to 6:00 Opening Reception (University Club)
We are grateful to Gale Cengage for its sponsorship.

Tuesday, 5 June 2018

9:00 to Noon Classes in Session

12:15 to 1:15 Lunch break / Unconference
“Mystery” Lunches
- DHSI Lunchtime Workshop Session (click for workshop details and free registration for DHSI participants)
  - 73. Introduction to ORCID (Digital Scholarship Commons, Classroom).

1:30 to 4:00 Classes in Session

4:15 to 5:15 DHSI Colloquium Lightning Talk Session 1 (MacLaurin A144)
Chair: James O’Sullivan
- New Modes of DH and Archival Skills Acquisition in a Graduate Public History Course. Paulina Rousseau (Ryerson U)
- Walking a Transect: Exploring a Soundscape. John Barber (Washington State U)
- Centering the Edge Case: Designing Services for Humanities Data Research. Grace Afsari-Mamagani (New York U)
- Orwellian Vocabulary and the 21st-Century Politics. Ilgin Kizilgunesler (U Manitoba)
- Making Open Data from a Gray Archive. Sara Palmer (Emory U)

6:00 to 8:00 DHSI Newcomer’s Beer-B-Q (Felicitas, Student Union Building)

Wednesday, 6 June 2018

9:00 to Noon Classes in Session

Lunch break / Unconference
“Mystery” Lunches
- Brown Bag Lecture: Alexandra Branzan Albu (U Victoria): “Visual Recognition of Symbolic and Natural Patterns” (Digital Scholarship Commons, 3rd Floor McPherson Library)

Abstract: Image-based object recognition is a visual pattern recognition problem; one may characterize visual patterns as either symbolic or natural. Symbolic patterns evolved for human communication; they include but are not limited to text, forms, tables, graphics, engineering drawings etc. Symbolic patterns vary widely in terms of size, style, language, alphabet and fonts; however, literate humans can easily compensate for this variability and instantly recognize most symbolic patterns. On the other hand, natural patterns characterize images of physical structures; they often lack the intrinsic discriminability and structure of symbolic patterns, and vary widely in terms of pose, perspective, and lighting.

This lecture will explore similarities and differences in approaches designed for recognizing visual and symbolic patterns, and will address the following questions via examples.
- What are the distinctive characteristics of natural patterns? What dimensions of variability can we infer?
- What are the distinctive characteristics of symbolic patterns? What dimensions of variability can we infer?

Alexandra Branzan Albu is an Associate Professor with the Department of Electrical and Computer Engineering and cross-listed with Computer Science. Her research interests are related to image analysis, computer vision, and visual computing. She is actively pursuing outreach activities dedicated to increasing the women's presence in electrical engineering and computer science.
Thursday, 7 June 2018

9:00 to Noon Classes in Session

12:15 to 1:15 UVIC Library/ETCL lunchtime talk: “A Humanities Application of 3D printing and Machine Translation in the ChessBard and Loss Sets” by Dr. Aaron Tucker

Thursday, 7 June 2018

4:15 to 5:15 DHSI Colloquium Lightning Talk Session 2 (MacLaurin A144)
Chair: James O’Sullivan

- Defining a Taxonomy of of Abandonment for Online Digital Humanities Projects. Luis Meneses (Electronic Textual Cultures Lab, U Victoria) and Jonathan Martin (King’s College London)
- The Stories We Tell: Representing Gay and Lesbian History through Digital Technologies in the LGLC Project. Nadine Boulay (Simon Fraser University) and Ewan Matthews (Ryerson U)
- Italian Paleography in the Digital Domain. Isabella Magni (Newberry Library)
- Digital Humanities, A Question of Ethics. Negar Basiri (Louisiana State U)
- Writing Poetry in High School. Guadalupe Echegoyen (National Autonomous U Mexico)

6:00 to 7:00 “Half Way There!” [An Informal, Self-Organized Birds of a Feather Get-Together] (Felicitas, Student Union Building)
Bring your DHSI nametag and enjoy your first tipple on us!

Friday, 8 June 2018 [DHSI; DLFxDHSI Opening]

9:00 to Noon DHSI Classes in Session

12:15 to 1:15 DHSI Lunch Reception / Course E-Exhibits (MacLaurin A100)

1:00 to 2:00 DLFxDHSI Registration (MacLaurin A100)

1:30 to 1:50 [DHSI] Remarks, A Week in Review (MacLaurin A144)

2:00 to 3:00 Joint Institute Lecture (DHSI and DLFxDHSI):
Bethany Nowviskie (CLIR DLF and U Virginia): “Reconstitute the World: Machine-reading Archives of Mass Extinction”
Chair: Lisa Goddard (U Victoria) (MacLaurin A144)

Abstract: The basic constitution of our digital collections becomes vastly more important in the face of two understandings: first, that archives of modernity are archives of the sixth great mass extinction of life on our planet; and next, that we no longer steward cultural heritage for human readers alone. In the same way that we people are shaped by what we read, hear, and see, the machine readers that follow us into and perhaps beyond the Anthropocene have begun to learn from “unsupervised” encounters with our digital libraries. What will we preserve for the living generations and artificial intelligences that will come? What do we neglect, or even choose to extinguish? And from an elegiac archive, a library of endings, can we create forward-looking, speculative collections—collections from which to deep-dream new futures? The most extra/ordinary power we possess is the power to make poetry from records of the past. Could it be called on, one day, to reconstitute the world?
3:30 to 5:00

Joint Reception: DHSI and DLFxDHSI (University Club)
DLFxDHSI Poster/Demo Session
- DHSI Colloquium Poster/Demo Session
  - Mediars as a Colonialist Artifact in Menzies' Journal. Paula Johanson (U Victoria)
  - Camp Edit: the Institute for the Editing of Historical Documents. NikolausWasmoen (Association for Documentary Editing, U Buffalo), Jennifer Stertzner (Association for Documentary Editing, U Virginia), and Cathy Moran Hajo (Association for Documentary Editing, Ramapo C)
  - A Digital Archaeology of Life in Cleveland's Depression-Era Slums. Charlie Harper (Case Western Reserve U) and Jared Bendis (Case Western Reserve U)
  - Feminist Pest Control: controlling and not controlling nonhuman pests. Lindsay Garcia (C of William and Mary)
  - Legends of the Buddhist Saints. Jonathan S. Walters (Whitman C) and Dana Johnson (Freelance Web Developer)
  - Accessibility in Digital Environments Via TEI-Encoded Uncontracted Braille. Gia Alexander (Texas A&M U)
  - Translation3point0: Why Literary Translation Data Matters. Katie King (U Washington)
  - PoéticaSonora: A Digital Audio Repository Prototype for Latin American Sound Art and Poetry. Aurelio Meza (Concordia U)
  - Beauty and the Book: Pre-Raphaelite Artistic Practice Contained. Josie Greenhill (U Victoria)
  - Poetic Procedures/Digital Deformances. Corey Sparks (California State U, Chico)
  - Miranda, the Folger Shakespeare Library's new Digital Asset Platform. Meaghan Brown (Folger Shakespeare Library)
  - Living Song Project. Quinn Patrick Ankrum (U Cincinnati) and Elizabeth Avery (U Oklahoma)
  - Digital Frankenstein Variorum. Rikk Mulligan (Carnegie Mellon U)

Saturday, 9 June 2018 [DLFxDHSI + DHSI Conference and Colloquium]

8:30 to 9:00
DLFxDHSI Registration (MacLaurin A100)

9:00 to 5:00
DLFxDHSI UnConference Sessions
- DHSI All Day Workshop Session (click for workshop details and free registration for DHSI participants)

9:00 to 4:00
- 53. Building Your Academic Digital Identity (MacLaurin D105, Classroom)

9:00 to 4:00
- DHSI Colloquium Day Conference (MacLaurin A144)

Welcome

People I: Documenting Online Lives. Chair: Molly Nebiolo (University of New York)
- Examining Gendered Harassment Online and in Silicon Valley. Andrea Flores (Ulta College)
- This is Just to Say I Have <X> the <Y> in your <Z>: Modernist Memes in an Era of Public Apology. Shawna Ross (Texas A&M University)

Break

People II: Documenting Lives Online. Chair: Dheepa Sundaram (College of Wooster)
- Youtube Yoga and Ritual on Demand: The Virtual Economics of Hindu Soteriology. Dheepa Sundaram (College of Wooster)
- The Resemblage Project: Creativity and Digital Health Humanities in Canada. Andrea Charise (University of Toronto) and Stefan Krecsy (University of Toronto)

Lunch

Projects I: Building and Analyzing. Chair: Yannis Rammos (New York University)
- Building the ARTECHNE Database: New directions in Digital Art History. Marieke Hendriksen (Old Dominion University)
- The Ineffective Inquisition: The Holy Office's Sphere of Influence in Early Modern New Spain. Kira Homo (Pennsylvania State University)

Break

Projects II: Mapping and Visualizing. Chair: Innocent Opara (Quomet Institute)
- Mapping Sarah Sophia Bank's Numismatic Collection. Erica Hayes (North Carolina State University) and Kacie Wills (University of California, Riverside)
- Text Mining and Visualizing 18th Century American Correspondence. Ashley Sanders Garcia (University of California, Los Angeles)

Break

Practices: Digital Scholarship on Campus and in the Classroom. Chair: Alyssa Arhuckle (University of Victoria)
### Sunday, 10 June 2018 [SINM + DHSI Registration, Workshops]

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:30 to 9:00</td>
<td>Symposium on Indigenous New Media Registration</td>
<td>MacLaurin A100</td>
</tr>
<tr>
<td>9:00 to 5:00</td>
<td>DHSI Registration</td>
<td>MacLaurin A100</td>
</tr>
<tr>
<td>9:00 to 4:00</td>
<td>63. Symposium on Indigenous New Media: Reading Group (Hickman 105, Classroom)</td>
<td></td>
</tr>
<tr>
<td>9:00 to 4:00</td>
<td>72. Symposium on Indigenous New Media: Indigitization (Hickman 120, Classroom)</td>
<td></td>
</tr>
<tr>
<td>9:00 to 4:00</td>
<td>DHSI All Day Workshop Sessions (click for workshop details and free registration for DHSI participants)</td>
<td></td>
</tr>
<tr>
<td>9:00 to 4:00</td>
<td>53. Building Your Academic Digital Identity (MacLaurin D105, Classroom)</td>
<td></td>
</tr>
<tr>
<td>9:00 to 4:00</td>
<td>54. An Introduction to the Archaeology of 1980s Computing (MacLaurin D114, Classroom)</td>
<td></td>
</tr>
<tr>
<td>9:00 to Noon</td>
<td>DHSI AM Workshop Sessions (click for workshop details and free registration for DHSI participants)</td>
<td></td>
</tr>
<tr>
<td>9:00 to Noon</td>
<td>55. Regular Expressions (MacLaurin D111, Classroom)</td>
<td></td>
</tr>
<tr>
<td>9:00 to Noon</td>
<td>56. 3D Visualization for the Humanities (MacLaurin D010, Classroom)</td>
<td></td>
</tr>
<tr>
<td>9:00 to Noon</td>
<td>58. DH Fieldwork Methods (MacLaurin D016, Classroom)</td>
<td></td>
</tr>
<tr>
<td>9:00 to Noon</td>
<td>60. Pedagogy of the Digitally Oppressed: Inculcating De-/Anti-/Post-Colonial Digital Humanities (MacLaurin D107, Classroom)</td>
<td></td>
</tr>
<tr>
<td>9:00 to Noon</td>
<td>61. Introduction to #GraphPoem. Digital Tools for Poetry Computational Analysis and Graph Theory Apps in Poetry (MacLaurin D101, Classroom)</td>
<td></td>
</tr>
<tr>
<td>9:00 to Noon</td>
<td>62. Creating a CV for Digital Humanities Makers (MacLaurin D115, Classroom)</td>
<td></td>
</tr>
<tr>
<td>1:00 to 4:00</td>
<td>DHSI PM Workshop Sessions (click for workshop details and free registration for DHSI participants)</td>
<td></td>
</tr>
<tr>
<td>1:00 to 4:00</td>
<td>64. Agent-Based Modelling in the Humanities (MacLaurin D111, Classroom)</td>
<td></td>
</tr>
<tr>
<td>1:00 to 4:00</td>
<td>65. Unleash Linux on MacOS (MacLaurin D010, Classroom)</td>
<td></td>
</tr>
<tr>
<td>1:00 to 4:00</td>
<td>66. DHSI Knits: History of Textiles and Technology (MacLaurin D016, Classroom)</td>
<td></td>
</tr>
<tr>
<td>1:00 to 4:00</td>
<td>67. Crowdsourcing as a Tool for Research and Public Engagement (MacLaurin D109, Classroom)</td>
<td></td>
</tr>
<tr>
<td>1:00 to 4:00</td>
<td>69. Web Annotation as Critical Humanities Practice (MacLaurin D103, Classroom)</td>
<td></td>
</tr>
<tr>
<td>1:00 to 4:00</td>
<td>70. Dynamic Ontologies for the Humanities (MacLaurin D107, Classroom)</td>
<td></td>
</tr>
<tr>
<td>1:00 to 4:00</td>
<td>71. Social Media Research in the Humanities (MacLaurin D109, Classroom)</td>
<td></td>
</tr>
<tr>
<td>4:10 to 5:00</td>
<td>Joint Institute Lecture (DHSI and SINM):</td>
<td></td>
</tr>
<tr>
<td>4:10 to 5:00</td>
<td>David Gaertner (U British Columbia): &quot;A Landless Territory?: CyberPowWow and the Politics of Indigenous New Media.&quot;</td>
<td></td>
</tr>
<tr>
<td>4:10 to 5:00</td>
<td>Chair: Deanna Reder (Simon Fraser U)</td>
<td></td>
</tr>
<tr>
<td>4:10 to 5:00</td>
<td>Abstract: Following the 1997 launch of Skawennati's (Mohawk) CyberPowWow, digital space has become a vital new territory for the resurgence of indigenous storytelling and cultural practice: &quot;We have signed a new treaty,&quot; Cree artist Archer Pechawis wrote of this period, &quot;and it is good. We have the right to hunt, fish, dance and make art at <a href="http://www.CyberPowWow.net">www.CyberPowWow.net</a>, .org and .com for as long as the grass grows and the rivers flow.&quot; This talk will critically explore the theoretical, cultural, political-economic, and gendered dynamics underwriting the histories and futures of Indigenous new media. Particular attention will be given in examining the ways in which new media and digital storytelling connect to and support key issues in the field of Indigenous studies, such as sovereignty, self-determination, decolonization, and land rights.</td>
<td></td>
</tr>
</tbody>
</table>

After the day, many will wander to Cadboro Bay and the pub at Smuggler's Cove OR the other direction to Shelbourne Plaza and Maude Hunter’s Pub OR even into the city for a bite to eat.
Your hosts for the week are Ray Siemens and Dan Sondheim.

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:45 to 8:15</td>
<td>DHSI Last-minute Registration (MacLaurin A100)</td>
<td></td>
</tr>
<tr>
<td>8:30 to 10:00</td>
<td>DHSI Welcome, Orientation, and Instructor Overview (MacLaurin A144)</td>
<td></td>
</tr>
<tr>
<td>9:00 to 4:00</td>
<td>SINM Sessions</td>
<td></td>
</tr>
<tr>
<td>10:15 to Noon</td>
<td>DHSI Classes in Session (click for details and locations)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 27. [Foundations] Understanding The Predigital Book: Technology and Texts (McPherson Library A003, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 28. [Foundations] Developing a Digital Project (With Omeka) (Clearihue D132, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 29. [Foundations] Models for DH at Liberal Arts Colleges (&amp; 4 yr Institutions) (MacLaurin D109, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 32. Stylometry with R: Computer-Assisted Analysis of Literary Texts (Clearihue A102, Lab)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 33. Digital Storytelling (MacLaurin D111, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 34. Text Mapping as Modelling (Clearihue D131, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 35. Geographical Information Systems in the Digital Humanities (Clearihue A105, Lab)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 36. Open Access and Open Social Scholarship (MacLaurin D114, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 37. Introduction to Machine Learning in the Digital Humanities (Cornett A229, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 38. Queer Digital Humanities: Intersections, Interrogations, Iterations (MacLaurin D110, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 41. Using Fedora Commons / Islandora (Human and Social Development A160, Lab)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 42. Documenting Born Digital Creative and Scholarly Works for Access and Preservation (MacLaurin D115, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 43. Games for Digital Humanists (MacLaurin D016, Classroom &amp; Human and Social Development A170, Lab)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 44. XPath for Document Archeology and Project Management (Cornett A128, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 45. Surveillance and the Digital Humanities (MacLaurin D103, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 46. Text Analysis with Python and the Natural Language ToolKit (Clearihue A103, Lab)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 47. Information Security for Digital Researchers (Clearihue D130, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 48. Wrangling Big Data for DH (Human and Social Development A150, Lab)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 49. Accessibility &amp; Digital Environments (MacLaurin D101, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 50. Critical Pedagogy and Digital Praxis in the Humanities (MacLaurin D105, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 51. Drupal for Digital Humanities Projects (MacLaurin D107, Classroom)</td>
<td></td>
</tr>
<tr>
<td>12:15 to 1:15</td>
<td>Lunch break / Unconference Coordination Session (MacLaurin A144)</td>
<td>MacLaurin A144</td>
</tr>
<tr>
<td></td>
<td>(Grab a sandwich and come on down!)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DHSI Undergraduate Meet-up, Brown-Bag (details via email)</td>
<td></td>
</tr>
<tr>
<td>1:30 to 4:00</td>
<td>DHSI Classes in Session</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ Joint Institute Lecture (DHSI and SINM):</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jordan Abel (Simon Fraser U): “Indigeneity, Conceptualism, and the Borders of DH.”</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chair: Michelle Brown (U Hawaii)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(MacLaurin A144)</td>
<td></td>
</tr>
<tr>
<td>4:10 to 5:00</td>
<td>Lunch break / Unconference</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“Mystery” Lunches</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ DHSI Lunchtime Workshop Session (click for workshop details and free registration for DHSI participants)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ 73. Introduction to ORCID (Digital Scholarship Commons, Classroom)</td>
<td></td>
</tr>
<tr>
<td>5:00 to 6:00</td>
<td>Joint Reception: DHSI and SINM (University Club)</td>
<td></td>
</tr>
<tr>
<td>9:00 to Noon</td>
<td>Classes in Session</td>
<td></td>
</tr>
<tr>
<td>12:15 to 1:15</td>
<td>Lunch break / Unconference</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“Mystery” Lunches</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ DHSI Lunchtime Workshop Session (click for workshop details and free registration for DHSI participants)</td>
<td></td>
</tr>
</tbody>
</table>

Tuesday, 12 June 2018
1:30 to 4:00  Classes in Session

4:15 to 5:15  DHSI Colloquium Lightning Talk Session 4 (MacLaurin A144)
Chair: Lindsey Seatter
- Mapping Indigenous and Chicana/o Environmental Imaginaries using GIS. Stevie Ruiz (California State U, Northridge), Quetzalli Enrique (California State U, Northridge), Enrique Ramirez (California State U, Northridge), and Tomas Figueroa (California State U, Northridge)
- Doing DH with Graphic Narratives. John Barber (Washington State U)
- "But is it any good?": A quantitative approach to the popularity of digital fanfiction. Suzanne Black (U Edinburgh)
- The American Prison Writing Archive (APWA). Doran Larson (Hamilton C), Janet Simons (Digital Humanities Initiative, Hamilton C), and William Rasenberger (Hamilton C)

6:00 to 8:00  DHSI Newcomer's Beer-B-Q (Felicitas, Student Union Building)

9:00 to Noon  Classes in Session

12:15 to 1:15  Lunch break / Unconference
*Mystery* Lunches

1:30 to 4:00  Classes in Session

4:15 to 5:15  DHSI Colloquium Lightning Talk Session 5 (MacLaurin A144)
Chair: Lindsey Seatter
- Faraway, so close: Has the political environment really changed in Ecuador?. Luis Meneses (Electronic Textual Cultures Lab, U Victoria)
- Re-mixing Melville's Reading: Text Analysis of Marginalia with R and XSLT. Christopher Ohge (U London, School of Advanced Study) and Steven Olsen-Smith (Boise State U)
- Developing Interactive and Open-Source OER: Inquiry-Based Music Theory. Evan Williamson (U Idaho)
- Spatial Humanities and the Web of Everywhere. Ken Cooper (SUNY Geneseo)

6:00 to 7:00  "Half Way There (yet again)!" [An Informal, Self-Organized Birds of a Feather Get-Together] (Felicitas, Student Union Building)
Bring your DHSI nametag and enjoy your first tipple on us!

9:00 to Noon  Classes in Session

12:15 to 1:15  Lunch Reception / Course E-Exhibits (MacLaurin A100)

1:30 to 4:00  Classes in Session

4:15 to 5:15  DHSI Colloquium Lightning Talk Session 6 (MacLaurin A144)
Chair: Lindsey Seatter
- Composition not Inheritance: Imagining a Functional Digital Humanities. Andrew Pilsch (Texas A&M U)
- Plotting Our Trajectories: Navigating, Situating, and Re-Inventing Research Topoi with R. Sean McCullough (Texas Christian University) and Jongkeyong Kim (Texas Christian U)
- Herb Simon and His Books. Avery Wiscomb (Carnegie Mellon U) and Daniel Evans (Carnegie Mellon U)
- (De/Re)Defining "The Digital": A Decolonial Approach to Digital Humanities. Ashley Caranto Morford (U Toronto) and Arun Jacob (McMaster U)

7:30 to 9:30  (Groovier?) Movie(r) Night (MacLaurin A144)

9:00 to Noon  Classes in Session

12:15 to 1:15  Lunch Reception / Course E-Exhibits (MacLaurin A100)
1:30 to 2:30

Institute Lecture: William Bowen (U Toronto Scarborough): “Discovery, Collaboration and Dissemination: Lessons Learned and Plans for the Future” (MacLaurin A144)

Abstract: Much has changed and continues to change in digital humanities since the formal establishment of Iter in the Fall of 1997. However, the mandate of the not-for-profit partnership to support “the advancement of learning in the study and teaching of Middle Ages and Renaissance (400–1700) through the development and distribution of online resources” continues to have relevance. This presentation explores the striking challenges faced by Iter and presents our current thinking on the realization of this mandate for the future through a platform with a focus on facilitating the discovery of the academic resources necessary to our work; creating an environment for collaboration, sharing and developing projects; and on enabling the distribution and publication of our scholarship.

2:40 to 3:00

Awards and Bursaries Recognition
Closing, DHSI in Review (MacLaurin A144)

Contact info:
institut@uvic.ca  P: 250-472-5401  F: 250-472-5681
DHSI 2018 -- Designing a Digital Project (with Omeka) Class Agenda

Monday, June 11

10:15-Noon: Class Introduction and intro to the digital humanities research process
Topics
Why and what do humanists need to know about conceptualizing a project?
What is design thinking and why should it matter to humanists?

Exercise: (Re)designing the humanities research experience

1:30-4: Introduction to the 3 aspects of a digital research project
Topics
The 3 aspects of a digital project
Approaching a digital project
Assessing technology affordances

Exercise: Assessing digital humanities projects for the 3 aspects of a digital project

Tuesday, June 12

10:15-Noon: Introduction to the 3 stages of a digital project
Topics
The 3 stages of a digital project
The influence of planning and one’s defining question
Common collection and analysis methods in DH projects

Exercise: Reverse engineer a digital project

1:30-4: Introduction to and working with Omeka
Topics
Omeka and what it can do
Omeka items, collections, and exhibits
Omeka themes
Omeka plugins (part 1)

Exercise: Create first item, collection and exhibit on Omeka.net

Wednesday, June 13
10:15-Noon: More advanced Omeka Topics
Omeka affordances
Omeka plugins (part 2)
Omeka + Neatline + other technology needed

Exercise: Collaboratively create a Neatline exhibit; practice with the Geolocation plugin

1:30-4: Design your research process (you can do a collaborative project if desired)

Exercise: Evaluate digital archive projects for the 3s of a digital project
Brainstorm project ideas

Thursday, June 14

10:15-Noon: Finish designing projects and start building with Omeka

Exercise: Continue with instructors and classmates to map out an Omeka project using what we learned about the 3s of digital projects.

1:30-4: Continue work with Omeka
Note
Instructors will be in assigned classroom and available to consult and aid in the coding process

Friday, June 15

10:15-Noon: Finish Omeka projects and Course review/reflection
Note
Instructors will be in assigned classroom and available to consult and aid in the coding process
Introduction

How does one approach a digital project?

Why are you in this class and why are we talking about conceptualizing Humanities technology projects? Chances are you’ve heard or seen a project from a peer or colleague, or maybe you have thought of, or even begun, a project of your own. Regardless of where you are in the process of a digital project, you are most likely at the mercy of others’ opinions when it comes to choosing how to proceed and from no fault of your own.

Though our goal at DHSI is not to make all, or even any of you, future humanist programmers, we do hope this class provides you a way to approach digital projects more comfortably and collaborate with others on them more successfully. Think of yourselves as Humanities scientists in training. This class will offer some training in learning the best methods for scoping digital projects, breaking down the process of the project (aka problem solving), and choosing the appropriate technology. But before we delve into these things, let’s answer a more basic question: why should humanists learn about creating digital projects?

Why should humanists learn about creating digital projects?

Whether we like it or not, we are in the digital age. If the materials that we are working with in our research are not born digital, making them digital or interacting with them digitally is becoming more and more the norm in academia. Digital historian Ben Schmidt clearly sets out the problems that arise from this change in the materials scholars interact with on his Sapping Attention blog:

Facing a digital source base with primarily expertise in close reading and navigating traditional archives, we are—whether we admit it or not—largely disarmed. A historian whose access is mediated by an archivist tends to know how best to interpret her sources; one plugging at databases through dimly-understood methods has lost his claim to expertise.

Schmidt suggests that not only do scholars today have the challenge of finding digital materials and deducing those that are most appropriate for one’s scholarship, but also deciphering the value and meaning of these sources. Archives have historically made all these steps easier. Though scholars would have to search the archive, the fact that the materials being hunted for were in a place limited one’s search and simultaneously proved the object’s value. If these materials are in an archive, they must be important to scholarship. But in the digital age, we
have potentially useful information all over the internet, and social media has completely changed who can participate in making history. Add to that all the hidden archival collections from lesser-known historical figures being made publically available on the web, and the scholar is left to determine the value of these new materials and if and how these materials can be incorporated into one’s research.

Digital technology and the web have not changed the three steps of the research process—data collection, analysis, and publication— they have merely impinged upon, and to an extent complicated, each stage. Your defining question guiding your research (aka, your thesis) does not necessarily change over the course of the project, but it does have an effect on each of these stages, affecting the way you collect and analyze data and the tools you use during each step of the way. For this reason, breaking a digital project down in stages makes it more manageable and achievable. A knowledge of how digital projects work under the hood will give you a leg up in determining when and what technologies to use at any given step of the research process.

Though this class is mainly an introduction to understanding the process of creating a digital project, it also serves as an introduction to using technology in your research. We will cover the basics of how to scope a digital project properly, but we will also discuss when and where to use certain technologies, as well as give you some hands on experience with the process using Omeka.

Over the course of the week, we will apply this kind of design thinking to our research and get some hands-on experience with prototyping and taking iterative steps to completing a digital project. Our goals for this course are for all of you to understand how and why you might use technology in your research process and how to break that process down into more manageable parts. We’ll call this developing a just enough, just in time research method.

Design Thinking and Digital Humanities

One of the toughest parts about getting started with digital projects is that you have to throw out a core feature of Humanities research: working alone and not disclosing what your research is about until you think it is ready. When doing a digital project you want to be more collaborative and get your ideas, in any manifestation, out as soon as possible and as often as possible. Digital projects often involve a level of “translation” and audience interaction that is quite foreign to traditional Humanities research, which makes getting peer and potential user feedback/participation early and often very important to the final success of the project. Thus, before actually doing a digital project, it’s best to learn this new research approach. But how? Enter design thinking.
We know, we know, let the buzz words start to fly. But honestly, the best way to a successful digital project is first to rethink the Humanities research experience. But before we can do that, we need to understand what is design thinking.

Design thinking, sometimes referred to as human-centered design, is a multi-stage, often iterative, process for finding a solution to a problem. There are five stages to the design thinking process:

1) **Empathize** - put yourself in your user's shoes through observation and interaction with him/her.
2) **Define** - through your observation, come up with a statement of the user's problem.
3) **Ideate** - explore all possible solutions for user's problem.
4) **Prototype** - transform one or more of your ideas into physical form
5) **Test** - get feedback from user and refine your solution.

In each stage, you learn more about whether or not your solution will work and, if needed, how to make your solution better. Let's get some practice with design thinking by re-imaging a classmate's Humanities research experience.

Part of the reason for rethinking the Humanities research process is so that we can step back and see the many parts of that process. When you have been trained for many years to do something and you have done it over and over again, oftentimes you begin to take the process for granted and think of it as one whole entity. When something foreign, like technology, gets introduced to the process, this kind of thinking can be very detrimental to the success of the digital project. Design thinking not only reminds us to tackle our project in stages, but also that there are many emotional ups and downs in this approach. To the left is an image from Tim Brown’s *Change By Design*, in which he borrows a diagram drawn by one of his fellow designers at IDEO to describe the emotional roller coaster of trying to design a new product. Does this inverse bell curve look familiar? Does your mapping of the typical Humanities research process look similar? Whether or not the answer to either question is true, this diagram shows us...
that when we feel worse about where we are in the project is when the most insight takes place. This is key to keep in mind when working on a digital project. In those times when you are most overwhelmed because your dataset is so large and/or diverse and you feel like there are a million different technologies that can be used on your project, is when the most insight can happen. Again, a main goal of this course is to give you experience working with technology in the context of your research so that you can begin to develop comfort and confidence in this process.
Conceptualizing a digital project

Like any good Humanities research, a digital project starts with a fleshed-out research question. Having a well-thought-out question becomes that much more important in digital projects because they can reach a broader audience and utilize various means to communicate. The digital scholar has to be more conscious of one's defining question and ensure that it is properly shaping every aspect and stage of the process. It is good to separate the digital research project into these two parts: aspects and stages. This allows the researcher to compartmentalize, for lack of better terms, one's goals and the research process. Let's dive into what these concepts mean and why it is good to keep them separate when doing digital research.

The 3 aspects of a digital project

What is an aspect? An aspect is not only a feature of the digital research project, but also a positioning of one's research goals. When doing any research, one has to consider the purpose and audience for the project. In traditional scholarship there is often a standard purpose and audience: a journal or book and your academic peers. From time to time, a scholar will write a book for a more popular audience, which will directly affect the purpose of the research. Such is the case too with a digital project, except that the audience and purpose are pretty limitless. Throw on top of these many purposes and audiences the various technologies that can be used to build the project, and you have the 3 aspects of a digital project. These aspects to a digital project are best represented in a Venn Diagram, where the overlap between these aspects represents a properly scoped project.
A better way to put this might be to call this your source + method. The material you use in your research (we’ll call it data from here on out) has certain qualities about it, like year published, author, genre, etc. (we’ll call this metadata from here on out), and the information in the object as well. A researcher could create a digital project that includes all of an object’s metadata and does not interpret it, thus, describing the source with complete fidelity. But doing so would not help the researcher make an argument about the data. Developing a research question involves finding the needed balance between fidelity to the data and arguing something about it. The more you figure out what you want to say and how to say it, the less details about the source you pay attention to.

Questions to answer when asking “what for?”:
- What do I want to highlight about the data?
- How much of the data do I want to use?
- How do I want to present the data?
- What message do I want to communicate about the data?
- How do I want to organize the data?

Who for?

As Humanists I am sure you heard or asked this question at least a million times: “Who’s your audience?” When we write papers in the Humanities, we generally have a certain audience in mind depending on the genre. Many digital projects, however, are presented online in ways that
make them accessible to many kinds of audiences; this makes this question a little harder to answer. In some cases, reaching a specific audience outside of academia might even be a central goal of the project so a well-thought-out imagined audience is key to overcoming our limited training in reaching out to various audiences. Who you are trying to communicate with will have a direct effect on what parts of your data you present and how you present them.

**Questions to answer when asking “who for?”:**
- What is my audience like?
- Is there more than one audience?
- What does my audience know about the data?
- How interested is the audience in the data?
- How does the audience expect to engage with the data?

**How to build?**

This is a very general way to question what technology will be best for the project. But it is not as simple as picking a technology and running with it. One has to assess the technology’s affordances (fancy work for what the technology can allow the user to do) and weigh them against the resources (people and money available to learn and build with that technology) and time you have available. If it was only this simple. When it comes to a technology’s affordances, one has to evaluate what level of functionality a technology has out of the box (weird way of saying, what a technology allows a user to do without any special configuration, add-ons, or
customization—think Wordpress plugin), its potential to be extended by add-ons and customization, and the activity and size of the community that supports and updates the technology. Throw on top of this the need to determine the long-term resources to support the project’s technology, and one has what goes into the "how build" question.

Questions to answer when asking “how build?”:

- What are the technology’s design and affordances out of the box?
- Is the technology extensible (does it have plugins)?
- Is any customization needed?
- Are there resources for functional or design customization?
- How much time to complete the project?
- What are the long-term technology costs?

The 3 stages of a digital project

The 3 stages of the digital project are no different than the stages in a traditional Humanities research project. They include data collection (gathering one’s resources), data analysis (annotating and organizing those resources), and data publication (final presentation of resources). The real purpose of thinking about the 3 stages is getting a researcher to consider the 3 aspects of the digital project in each of these stages.
Another reason why it is useful to consider the stages of a research project when working with technology is because a digital project is not always a begin at the first step, linear process. Rather it is perfectly acceptable to begin a digital project from either end of the data collection or publication phase. Most of the time, researchers will approach a digital project inductively, wherein they start by looking at their data to find a thesis from patterns in it. Yet sometimes a researcher may see a digital project that fits their information perfectly, in which they will work backwards to deduce how their data might fit this same final presentation. In either case, how the researcher turns the information collected into data is a key stage in the digital research process.

Data Collection

In traditional Humanities research, data collection usually consists of finding passages in books, journals, newspaper, etc. and organizing them in way conducive to making an argument. This might entail putting the passages on flash cards or highlighting them with different colors as a means to organizing and classifying their meaning into theories or other key features, like when they were produced or in what style. All this is done in a digital project as well, but with the added step of having to perform translation. Translation is the process of turning unstructured information, something a human is really good at comprehending, into structure data, what computers are really good at reading. So how do we do this process?

A useful first step in translation is to answer these basic questions about one’s information: what, where, who, when, and how. This will start breaking down the information into pieces that can be entered into a spreadsheet. Once this is done, we can start to perform two more key steps, standardization and classification.

1) standardization - process of deciding upon one way of representing a piece of information that appears in the source in a number of different ways

2) classification - process of grouping together information (‘strings’) according to some theoretical, empirical or entirely arbitrary scheme

Take a look at this visualization, http://poly-graph.co/vocabulary.html, and use the three steps of translation to develop a data structure for the information in the visualization.
This is what the information looks like after we apply the first step of translation. Below the data is highlighted for parts that are standardized (in red) and those that are classified (in orange).

<table>
<thead>
<tr>
<th>id</th>
<th>rapper</th>
<th>region</th>
<th>wutang?</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>rakim</td>
<td>Rakim (including Eric B. &amp; Rakim)</td>
<td>east</td>
<td>nowutang</td>
<td>4,612</td>
</tr>
<tr>
<td>dmx</td>
<td>DMX</td>
<td>east</td>
<td>nowutang</td>
<td>3,214</td>
</tr>
<tr>
<td>too-short</td>
<td>Too Short</td>
<td>west</td>
<td>nowutang</td>
<td>3,391</td>
</tr>
<tr>
<td>gza</td>
<td>GZA (only solo albums)</td>
<td>east</td>
<td>wutang</td>
<td>6,426</td>
</tr>
<tr>
<td>drake</td>
<td>Drake</td>
<td>other</td>
<td>nowutang</td>
<td>3,522</td>
</tr>
<tr>
<td>bone-thugs</td>
<td>Bone Thugs-n-Harmony</td>
<td>midwest</td>
<td>nowutang</td>
<td>3,547</td>
</tr>
<tr>
<td>master-p</td>
<td>Master P</td>
<td>south</td>
<td>nowutang</td>
<td>3,612</td>
</tr>
</tbody>
</table>

(standardization = red; classification = orange)

Looking back at the visualization, one can see how these translation steps help the researcher decide on how to organize and what to highlight in the data (the analysis stage of the research...
process), as well as inform one’s use of technology and presentation of the data (the publication stage of the research process).

**Analysis**

As one sees above, the analysis stage really starts in the later parts of the translation process. Once a researcher starts to classify information, the analysis process has begun. In the rappers’ vocabulary project, the creator’s decision to separate each rapper by geographic location and membership in the wu-tang clan signals a level of interpretation in that it highlights certain parts of the data while deemphasizing others. The project starts out with a simple question: who is the most linguistically diverse rapper of all time. When the author provides toggle buttons for location and wu-tang affiliation, he is implying that these features of the data may help one qualify what attributes go into a linguistically diverse rapper. In this case, that affiliation with an area or group may have some correlation to the uniqueness of one’s word choice.

Analysis is all about highlighting and hiding parts of the data (notice that the revenue of albums sales was excluded from this dataset whereas info about Shakespeare and Melville vocab were included) in order to make one’s argument stronger. According to the author of the rapper’s vocabulary visualization, he used a specific “research methodology” for determining uniqueness of words (token analysis), did all he could to ensure variants of an already counted word were not included in the uniqueness count, and was meticulous in ensuring each artist had 35000 words. All this is to prove that he has the formula and best approach to determining a rapper’s vocabulary prowess, which has a direct affect on the author’s presentation method.

**Publication**

Choosing a publication platform really starts in the data collection stage and is in constant flux until analysis is close to finished. The first step of translation--asking who, what, why, when, and where about your information--is especially good for deciding what type of digital output is appropriate for your data and opens the door to really important exploratory analysis. Here’s a simple list of potential output types for a digital project and how they might map to certain parts of the data that you find most significant for your argument:

- **data visualization** - typically displays quantitative data and takes the form of a chart/graph
- **temporal** - typically displays passing of time in a linear fashion
- **geospatial** - typically locates objects (ie, people, places, organizations) on a map
- **dimensional** - typically displays object spatially, often in 3D
- **narrative** - typically tells a story and can take many forms (i.e., temporal, spatial, quantitative)
- **network** - typically displays connection between objects (i.e., people, places, organizations)
- **archival** - typically describes all aspects of objects

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>data vis</strong></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td><strong>temporal</strong></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td><strong>geospatial</strong></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td><strong>dimensional</strong></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td><strong>narrative</strong></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td><strong>network</strong></td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td><strong>archival</strong></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

This mapping is not meant to be prescriptive as much as a starting point for efficiently deducing what potential communication formats your thesis-shaped dataset fits best. In the case of the Rappers’ Vocabulary, the what (uniqueness of language) is the most important piece, so off the bat you can eliminate temporal and geospatial representations because those will focus more on the when and where and detract from the focus on language use (throw out archival). The goal is not to tell a story about the most linguistically diverse rapper (throw out narrative) but show how all rappers stack up against each other (throw out network). Finally, we have no need to represent these rappers in 3D (throw out dimensional), so we are left with exploring the question about the most verbally unique rapper through a data visualization.

Try developing a narrative for how any of the following digital projects developed out of the three 3s - 3 aspects (what for? Who for? How to build?); 3 stages (collection, analysis, and publication); and 3 parts of translation (answering basic questions about one’s information, standardization, and classification):

- [https://podio.com/site/creative-routines](https://podio.com/site/creative-routines)
- [http://www.concerthotels.com/100-years-of-rock/](http://www.concerthotels.com/100-years-of-rock/)
- [http://cameronblevins.org/gotp/](http://cameronblevins.org/gotp/)
- http://www.slate.com/blogs/the_slatest/2015/10/06/syrian_conflict_relationships_explained.html
- https://britishmuseum.withgoogle.com/
Data Structure Exercise

Exercise: For each piece of information, create columns for each detail given about each event of gun violence. Then under the column put the corresponding details for each gun violence event.

Information 1

Birmingham police officer's father killed in shooting, 2 in custody
Posted: May 06, 2013 10:46 AM EDT
Updated: May 13, 2013 10:46 AM EDT
By Melynda Schauer

Birmingham police are investigating a homicide that happened Monday morning in the southwest part of the city. Police say a man was shot at 209 3rd Avenue SW near Zion Star Missionary Baptist Church around 9 a.m.

Police identify the victim as Birl Jackson, 76. Jackson was the father of a Birmingham police officer and had been a deacon at Mt. Calvary Baptist Church in West End for almost 40 years, according to members of his church. Jackson had been making a living cutting grass and maintaining lawns for decades. He was on the job with two of his grandsons when he was shot near his truck Monday morning.

A neighbor says he heard several gunshots and came outside to see the victim slumped over his truck. Jackson was rushed to UAB Hospital where he was pronounced dead.

Birmingham police think robbery may have been a motive in the shooting.

Information 2

18-year-old found dead in north Montgomery victim of city’s 42nd homicide

By Erin Edgemon
on October 23, 2013 at 10:25 AM, updated October 23, 2013 at 10:28 AM

Montgomery police is investigating the fatal shooting of 18-year Jevonte Scott as the city’s 42nd homicide of the year.
Scott was found dead of gunshot wounds in the backyard of a residence in the 100 block of Cedar Street at around 11 a.m. on Tuesday, Montgomery police released.
Father, baby son shot dead in Oakland
By Henry K. Lee, Justin Berton and Kurtis Alexander
Updated 11:20 pm, Wednesday, August 7, 2013

An 8-year-old girl at a sleepover. A pet sitter driving through her neighborhood. And now, a father and his 1-year-old son, in Oakland for a slain relative's funeral and sleeping in a bedroom filled with children.

The latest horror to visit a city already reeling from gun violence happened early Wednesday, when someone opened fire through the window of an East Oakland home where 17 members of an extended family were staying. When the shooting stopped, 20-year-old Andrew Thomas lay dying near his 1-year-old son, Drew Jackson. The boy, mortally wounded, died before sunrise at a hospital.

Unlike the July 17 killing of young Alaysha Carradine and the fatal shooting a week later of Judy Salamon in her car, police think someone set out to kill Thomas. But even longtime investigators were shocked that someone would fire indiscriminately into a bedroom where four children were sleeping.

"It's absolutely unacceptable that we have children being shot while they sleep in their bed," said interim Police Chief Sean Whent. "It is horrific."
Prototyping


Design Thinking


INTRODUCTION

the power of design thinking

an end to old ideas

Practically everyone who has visited England has experienced the Great Western Railway, the crowning achievement of the great Victorian engineer Isambard Kingdom Brunel. I grew up within earshot of the GWR, and as a child in rural Oxfordshire I often bicycled alongside the line and waited for the great express trains to roar past at more than one hundred miles an hour. The train ride is more comfortable today (the carriages now sport springs and cushioned seats) and the scenery has certainly changed, but a century and a half after it was built the GWR still stands as an icon of the industrial revolution—and as an example of the power of design to shape the world around us.

Although he was the engineer's engineer, Brunel was not solely interested in the technology behind his creations. While considering the design of the system, he insisted upon the flattest possible gradient because he wanted passengers to have the sense of “floating across the countryside.” He constructed bridges, viaducts, cuttings, and tunnels all in the cause of creating not just efficient transportation but the best possible expe-
rience. He even imagined an integrated transport system that would allow the traveler to board a train at London's Paddington Station and disembark from a steamship in New York. In every one of his great projects Brunel displayed a remarkable—and remarkably prescient—talent for balancing technical, commercial, and human considerations. He was not just a great engineer or a gifted designer; Isambard Kingdom Brunel was one of the earliest examples of a design thinker.

Since the completion of the Great Western Railway in 1841, industrialization has wrought incredible change. Technology has helped lift millions out of poverty and has improved the standard of living of a considerable portion of humanity. As we enter the twenty-first century, however, we are increasingly aware of the underside of the revolution that has transformed the way we live, work, and play. The sooty clouds of smoke that once darkened the skies over Manchester and Birmingham have changed the climate of the planet. The torrent of cheap goods that began to flow from their factories and workshops has fed into a culture of excess consumption and prodigious waste. The industrialization of agriculture has left us vulnerable to natural and man-made catastrophes. The innovative breakthroughs of the past have become the routine procedures of today as businesses in Shenzhen and Bangalore tap into the same management theories as those in Silicon Valley and Detroit and face the same downward spiral of commoditization.

Technology still has not run its course. The communications revolution sparked by the Internet has brought people closer together and given them the opportunity to share perspectives and create
new ideas as never before. The sciences of biology, chemistry, and physics have merged in the forms of biotechnology and nanotechnology to create the promise of lifesaving medicines and wondrous new materials. But these spectacular achievements are unlikely to help us reverse our ominous course. Just the opposite.

**We need new choices**

A purely technocentric view of innovation is less sustainable now than ever, and a management philosophy based only on selecting from existing strategies is likely to be overwhelmed by new developments at home or abroad. What we need are new choices—new products that balance the needs of individuals and of society as a whole; new ideas that tackle the global challenges of health, poverty, and education; new strategies that result in differences that matter and a sense of purpose that engages everyone affected by them. It is hard to imagine a time when the challenges we faced so vastly exceeded the creative resources we have brought to bear on them. Aspiring innovators may have attended a “brainstorming” session or learned a few gimmicks and tricks, but rarely do these temporary placeholders make it to the outside world in the form of new products, services, or strategies.

What we need is an approach to innovation that is powerful, effective, and broadly accessible, that can be integrated into all aspects of business and society, and that individuals and teams can use to generate breakthrough ideas that are implemented and that therefore have an impact. Design thinking, the subject of this book, offers just such an approach.
Design thinking begins with skills designers have learned over many decades in their quest to match human needs with available technical resources within the practical constraints of business. By integrating what is desirable from a human point of view with what is technologically feasible and economically viable, designers have been able to create the products we enjoy today. Design thinking takes the next step, which is to put these tools into the hands of people who may have never thought of themselves as designers and apply them to a vastly greater range of problems.

Design thinking taps into capacities we all have but that are overlooked by more conventional problem-solving practices. It is not only human-centered; it is deeply human in and of itself. Design thinking relies on our ability to be intuitive, to recognize patterns, to construct ideas that have emotional meaning as well as functionality, to express ourselves in media other than words or symbols. Nobody wants to run a business based on feeling, intuition, and inspiration, but an overreliance on the rational and the analytical can be just as dangerous. The integrated approach at the core of the design process suggests a “third way.”

swimming upstream

I was trained as an industrial designer, but it took me a long time to realize the difference between being a designer and thinking like a designer. Seven years of undergraduate and graduate education and fifteen years of professional practice went by before I had any real inkling that what I was doing was more than
n 2004 Shimano, a leading Japanese manufacturer of bicycle components, was experiencing flattening growth in its traditional high-end road racing and mountain bike segments in the United States. The company had always relied on new technology to drive its growth. It had invested heavily in an effort to anticipate the next innovation. In the face of the changing market it seemed prudent to try something new, so Shimano invited IDEO to collaborate.

What followed was an exercise in designer-client relations that looked very different from what such an engagement might have looked like a few decades or even a few years earlier. Shimano did not hand us a list of technical specifications and a binder full of market research and send us off to design a bunch of parts. Rather, we joined forces and set out together to explore the changing terrain of the cycling market.

During the initial phase, we fielded an interdisciplinary team of designers, behavioral scientists, marketers, and engineers whose task was to identify appropriate constraints for the project. The team began with a hunch that it should not focus on the high-end market. Instead, they fanned out to learn why 90 percent of American adults don’t ride bikes—despite the fact that 90 percent of them did as kids! Looking for new ways to think about the problem, they spent time with consumers from across the spec-
trum. They discovered that nearly everyone they met had happy memories of being a kid on a bike but many are deterred by cycling today—by the retail experience (including the intimidating, Lycra-clad athletes who serve as sales staff in most independent bike stores); by the bewildering complexity and excessive cost of the bikes, accessories, and specialized clothing; by the danger of cycling on roads not designed for bicycles; and by the demands of maintaining a sophisticated machine that might be ridden only on weekends. They noted that everyone they talked to seemed to have a bike in the garage with a flat tire or a broken cable.

This human-centered exploration—which looked for insights from bicycle aficionados but also, more important, from people outside Shimano's core customer base—led to the realization that a whole new category of bicycling might reconnect American consumers to their experiences as children. A huge, untapped market began to take shape before their eyes.

The design team, inspired by the old Schwinn coaster bikes that everyone seemed to remember, came up with the concept of "coasting." Coasting would entice lapsed bikers back into an activity that was simple, straightforward, healthy, and fun. Coasting bikes, built more for pleasure than for sport, would have no controls on the handlebars, no cables snaking along the frame, no nest of precision gears to be cleaned, adjusted, repaired, and replaced. As we remember from our earliest bikes, the brakes would be applied by backpedaling. Coasting bikes would feature comfortable padded seats, upright handlebars, and puncture-resistant tires and require almost no maintenance. But this is not simply a retrobike: it incorporates sophisticated engineering with an automatic transmission that shifts the gears as the bicycle gains speed or slows.
Three major manufacturers—Trek, Raleigh, and Giant—began to develop new bikes incorporating innovative components from Shimano, but the team didn’t stop there. Designers might have ended the project with the bike itself, but as holistic design thinkers they pressed ahead. They created in-store retailing strategies for independent bike dealers, in part to mitigate the discomfort that novices felt in retail settings built to serve enthusiasts. The team developed a brand that identified coasting as a way to enjoy life (“Chill. Explore. Dawdle. Lollygag. First one there’s a rotten egg.”). In collaboration with local governments and cycling organizations, it designed a public relations campaign including a Web site that identified safe places to ride.

Many other people and organizations became involved in the project as it passed from inspiration through ideation and on into the implementation phase. Remarkably, the first problem the designers would have been expected to address—the look of the bikes—was deferred to a late stage in the development process, when the team created a “reference design” to show what was possible and to inspire the bicycle manufacturers’ own design teams. Within a year of the bike’s successful launch, seven more manufacturers had signed up to produce coasting bikes. An exercise in design had become an exercise in design thinking.

**three spaces of innovation**

Although I would love to provide a simple, easy-to-follow recipe that would ensure that every project ends as successfully as
this one, the nature of design thinking makes that impossible. In contrast to the champions of scientific management at the beginning of the last century, design thinkers know that there is no "one best way" to move through the process. There are useful starting points and helpful landmarks along the way, but the continuum of innovation is best thought of as a system of overlapping spaces rather than a sequence of orderly steps. We can think of them as inspiration, the problem or opportunity that motivates the search for solutions; ideation, the process of generating, developing, and testing ideas; and implementation, the path that leads from the project room to the market. Projects may loop back through these spaces more than once as the team refines its ideas and explores new directions.

The reason for the iterative, nonlinear nature of the journey is not that design thinkers are disorganized or undisciplined but that design thinking is fundamentally an exploratory process; done right, it will invariably make unexpected discoveries along the way, and it would be foolish not to find out where they lead. Often these discoveries can be integrated into the ongoing process without disruption. At other times the discovery will motivate the team to revisit some of its most basic assumptions. While testing a prototype, for instance, consumers may provide us with insights that point to a more interesting, more promising, and potentially more profitable market opening up in front of us. Insights of this sort should inspire us to refine or rethink our assumptions rather than press onward in adherence to an original plan. To borrow the language of the computer industry, this approach should be seen not as a system reset but as a meaningful upgrade.

The risk of such an iterative approach is that it appears to
extend the time it takes to get an idea to market, but this is often a shortsighted perception. To the contrary, a team that understands what is happening will not feel bound to take the next logical step along an ultimately unproductive path. We have seen many projects killed by management because it became clear that the ideas were not good enough. When a project is terminated after months or even years, it can be devastating in terms of both money and morale. A nimble team of design thinkers will have been prototyping from day one and self-correcting along the way. As we say at IDEO, “Fail early to succeed sooner.”

Insofar as it is open-ended, open-minded, and iterative, a process fed by design thinking will feel chaotic to those experiencing it for the first time. But over the life of a project, it invariably comes to make sense and achieves results that differ markedly from the linear, milestone-based processes that define traditional business practices. In any case, predictability leads to boredom and boredom leads to the loss of talented people. It also leads to results that rivals find easy to copy. It is better to take an experimental approach: share processes, encourage the collective ownership of ideas, and enable teams to learn from one another.

A second way to think about the overlapping spaces of innovation is in terms of boundaries. To an artist in pursuit of beauty or a scientist in search of truth, the bounds of a project may appear as unwelcome constraints. But the mark of a designer, as the legendary Charles Eames said often, is a willing embrace of constraints.

Without constraints design cannot happen, and the best design—a precision medical device or emergency shelter for
disaster victims—is often carried out within quite severe constraints. For less extreme cases we need only look at Target’s success in bringing design within the reach of a broader population for significantly less cost than had previously been achieved. It is actually much more difficult for an accomplished designer such as Michael Graves to create a collection of low-cost kitchen implements or Isaac Mizrahi a line of ready-to-wear clothing than it is to design a teakettle that will sell in a museum store for hundreds of dollars or a dress that will sell in a boutique for thousands.

The willing and even enthusiastic acceptance of competing constraints is the foundation of design thinking. The first stage of the design process is often about discovering which constraints are important and establishing a framework for evaluating them. Constraints can best be visualized in terms of three overlapping criteria for successful ideas: feasibility (what is functionally possible within the foreseeable future); viability (what is likely to become part of a sustainable business model); and desirability (what makes sense to people and for people).

A competent designer will resolve each of these three constraints, but a design thinker will bring them into a harmonious balance. The popular Nintendo Wii is a good example of what happens when someone gets it right. For many years a veritable arms race of more sophisticated graphics and more expensive consoles has been driving the gaming industry. Nintendo realized that it would be possible to break out of this vicious circle—and create a more immersive experience—by using the new technology of gestural control. This meant less focus on the resolution of the screen graphics, which in turn led to a less expensive console and better margins on the product. The Wii
strikes a perfect balance of desirability, feasibility, and viability. It has created a more engaging user experience and generated huge profits for Nintendo.

This pursuit of peaceful coexistence does not imply that all constraints are created equal; a given project may be driven disproportionately by technology, budget, or a volatile mix of human factors. Different types of organizations may push one or another of them to the fore. Nor is it a simple linear process. Design teams will cycle back through all three considerations throughout the life of a project, but the emphasis on fundamental human needs—as distinct from fleeting or artificially manipulated desires—is what drives design thinking to depart from the status quo.
building to think, 
or the power of prototyping

Lego launched me on my career as a design thinker. In the early 1970s, when I was nine or ten, England was going through yet another of its periodic recessions and the coal miners had waited until winter to go out on strike. This meant no coal for the power stations, which meant not enough electricity to meet demand, which meant regular blackouts. Determined to do my bit, I marshaled my entire inventory of Legos and built a great big flashlight using some fancy light bricks that glowed in the dark. I proudly handed the flashlight to my mother so that she had enough light to cook my dinner. I had built my first prototype.

By the age of ten I had learned the power of prototyping based on years of intensive study. As a younger child I had spent hours using Legos and Meccano (known to Americans as Erector Sets) to create a world full of rocket ships, dinosaurs, and robots of every imaginable size and shape. Like every other kid, I was thinking with my hands, using physical props as a springboard for my imagination. This shift from physical to abstract and back again is one of the most fundamental processes by which we explore the universe, unlock our imaginations, and open our minds to new possibilities.

Most companies are full of people who have set aside such childish pursuits and moved on to more important matters such
as writing reports and filling out forms, but one thing strikes
the visitor to an organization that uses design thinking: as in
any child’s bedroom, there are prototypes everywhere. Peek in-
side a project room, and you will see prototypes on every sur-
face. Walk the halls, and you will see prototypes being used to
tell stories about past projects. You will see prototyping tools
ranging from X-acto knives and masking tape to $50,000 laser
cutters. Whatever the budget and whatever the facilities, pro-
totyping will be the essence of the place.

Frank Lloyd Wright claimed that his early childhood ex-
perience with Froebel kindergarten blocks—developed by
Friedrich Froebel in the 1830s to help children learn the prin-
ciples of geometry—ignited his creative passion: “The maple-
wood blocks . . . are in my fingers to this day,” he wrote in
his autobiography. Charles and Ray Eames, one of the greatest
prototyping teams of all times, used prototyping to explore and
refine ideas, sometimes over many years. The result was noth-
ing short of the reinvention of twentieth-century furniture.
As by a curious admirer whether the iconic Eames lounge
chair came to him in a flash, Charles replied, “Yes, sort of a
thirty-year flash.”

Since openness to experimentation is the lifeblood of any
creative organization, prototyping—the willingness to go
ahead and try something by building it—is the best evidence
of experimentation. We may think of a prototype as a fin-
ished model of a product about to be manufactured, but that
definition should be carried much further back in the pro-
cess. It needs to include studies that may appear rough and
simple and encompass more than just physical objects. Fur-
thermore, it’s not necessary to be an industrial designer to
adopt the habit of prototyping: financial services executives, retail merchants, hospital administrators, city planners, and transportation engineers can and should participate in this essential component of design thinking, as we shall see. David Kelley calls prototyping “thinking with your hands,” and he contrasts it with specification-led, planning-driven abstract thinking. Both have value and each has its place, but one is much more effective at creating new ideas and driving them forward.

**quick and dirty**

Although it might seem as though frittering away valuable time on sketches and models and simulations will slow work down, prototyping generates results faster. This seems counterintuitive: surely it takes longer to build an idea than to think one? Perhaps, but only for those gifted few who are able to think the right idea the first time. Most problems worth worrying about are complex, and a series of early experiments is often the best way to decide among competing directions. The faster we make our ideas tangible, the sooner we will be able to evaluate them, refine them, and zero in on the best solution.

Gyrus ACMI is on the cutting edge of surgical instrumentation and a leader in developing techniques for minimally invasive surgery. In 2001 IDEO began to work with Gyrus to develop a new apparatus for operating on delicate nasal tissues. Early on in the project the team met with six otolaryngology surgeons to learn how they performed the procedure, the problems with existing instruments, and what characteristics they
might be looking for in a new system. One of the surgeons, using imprecise words and awkward hand gestures, described how he might prefer a device with a pistol grip. After they departed one of our designers had grabbed a whiteboard marker and a 35 mm film canister and taped them to a plastic clothespin that was lying nearby, and squeezed the clothespin as if it were a trigger. This rudimentary prototype catapulted the discussion forward, put everyone on the same page, and saved countless meetings, videoconferences, shop time, and airplane tickets. Cost of the prototype in labor and materials: $0 (we were able to rescue the marking pen).

Just as it can accelerate the pace of a project, prototyping allows the exploration of many ideas in parallel. Early prototypes should be fast, rough, and cheap. The greater the investment in an idea, the more committed one becomes to it. Overinvestment in a refined prototype has two undesirable consequences: First, a mediocre idea may go too far toward realization—or even, in the worst case, all the way. Second, the prototyping process itself creates the opportunity to discover new and better ideas at minimal cost. Product designers can use cheap and easy-to-manipulate materials: cardboard, surfboard foam, wood, and even objects and materials they find lying around—anything they can glue or tape or staple together to create a physical approximation of ideas. IDEO’s first and greatest prototype was created when the company consisted of eight scruffy designers crowded together in a studio above Roxy’s dress shop on University Avenue in Palo Alto. Douglas Dayton and Jim Yurchenco affixed the roller ball from a tube of Ban Roll-on deodorant to the base of a plastic butter dish. Before long Apple Computer was shipping its first mouse.
enough is enough

Prototypes should command only as much time, effort, and investment as is necessary to generate useful feedback and drive an idea forward. The greater the complexity and expense, the more “finished” it is likely to seem and the less likely its creators will be to profit from constructive feedback—or even to listen to it. The goal of prototyping is not to create a working model. It is to give form to an idea to learn about its strengths and weaknesses and to identify new directions for the next generation of more detailed, more refined prototypes. A prototype’s scope should be limited. The purpose of early prototypes might be to understand whether an idea has functional value. Eventually designers need to take the prototype out into the world to get feedback from the intended users of the final product. At this point the surface qualities of the prototype may require a bit more attention so that potential consumers are not distracted by the rough edges or unresolved details. Most people, for example, will find it difficult to visualize how a washing machine made of cardboard will work.

Some pretty amazing technology is available today for designers to create prototypes quickly and at an extremely high level of fidelity, including ultraprecise laser cutters, computer-aided design tools, and machines that function as 3-D printers. Sometimes they are too good, as we discovered when a Steelcase executive, mistaking an expertly detailed foam model for the real thing, destroyed a $40,000 prototype of the Vecta chair by sitting on it. But all the technology in the world will come to naught if it is used to create prototypes too refined, too detailed, and too early. “Just enough prototyping” means
picking what we want to learn about and achieving just enough resolution to make that the focus. An experienced prototyper knows when to say “Enough is enough.”

prototyping things you can’t pick up

Most imaginable prototypes up to this point refer to physical products—stuff that hurts when you trip over it or drop it on your toes. The same rules apply when the challenge is a service, a virtual experience, or even an organizational system.

Anything tangible that lets us explore an idea, evaluate it, and push it forward is a prototype. I have seen sophisticated insulin injection devices that began life as Legos. I have seen software interfaces mocked up with Post-it notes long before a line of code was written. I have seen new concepts for neighborhood banking acted out before clients as a skit, against a backdrop of “counters” made of flimsy foam core—a kind of cardboard material that is very strong, very light, and very cheap—held together with masking tape. In each case an idea has been given expression through an appropriate medium to show to others for feedback.

The movie industry has long used this practice. Once upon a time, when film was little more than a recorded version of theater, it was feasible to go from a script straight to shooting the movie. But as directors grew more ambitious—and audiences more demanding—they began to include multiple cameras and special effects. The storyboard emerged as a way of mapping out the movie before it was shot to make sure that all the scenes were thought through and that the director wouldn’t get to the
editing room only to find a vital angle or crucial shot flawed or
missing. As filmmaking grew ever more sophisticated, espe-
cially pioneered by Walt Disney Studios' animation, the story-
board took on an even more important role. It became a pro-
totyping tool that enabled animators to assure themselves that
the story hung together before the detail work began. Today,
with sophisticated, expensive digital special effects dominating
so much of Hollywood, filmmakers have moved to computer-
based storyboards and "animatics" to test the motion in a shot
before they commit to the real thing.

Techniques borrowed from film and other creative indus-
tries suggest how we might prototype nonphysical experiences.
These include scenarios, a form of storytelling in which some
potential future situation or state is described using words and
pictures. We might, for example, invent a character who fits
a set of demographic factors that interest us—a divorced pro-
fessional woman with two small children, for instance—and
develop a believable scenario around her daily routine in order
to "observe" how she might use an electric vehicle charger or an
online pharmacy.

When Wi-Fi communications were in their infancy, Vo-
cera developed a video scenario to demonstrate how employees
might use a wearable, voice-controlled "communications badge"
to stay connected with coworkers anywhere within a company's
network. The short movie followed the rounds of a fictional
IT support team and was far more effective in explaining the
concept to potential investors than a technical brief or a
deck of PowerPoint slides. Sony used the same technique when
it was developing its first online concepts in the early 1990s. A
design team created scenarios around the lives of teenagers in
Tokyo to show how they might use new kinds of online gaming parlors to play interactive video games or sing karaoke songs together. In the early years of the Internet these plausible fictions helped management visualize how it might become the basis of new services and business models.

Another considerable value of scenarios is that they force us to keep people at the center of the idea, preventing us from getting lost in mechanical or aesthetic details. They remind us at every moment that we are not dealing with things but with what the psychologist Mihaly Csikszentmihalyi calls “transactions between people and things.” Prototyping at work is giving form to an idea, allowing us to learn from it, evaluate it against others, and improve upon it.

A simple scenario structure useful in the development of new services is the “customer journey.” This structure charts the stages through which an imagined customer passes from the beginning of a service experience to the end. The starting point may be imaginary, or it may come directly from observations of people purchasing an airline ticket or deciding whether or not to install solar panels on a roof. In either case, the value of describing a customer journey is that it clarifies where the customer and the service or brand interact. Every one of these “touchpoints” points to an opportunity to provide value to a firm’s intended customers—or to derail them for good.

Some years ago Amtrak began studying opportunities to improve transportation on the East Coast by offering a high-speed train service between Boston, New York, and Washington, D.C. By the time Amtrak invited IDEO to participate in what would become the Acela project, the focus had narrowed to the trains themselves and, in fact, to the design of the seats.
After spending countless days riding trains with customers, the team created a simple customer journey that described the entire travel process. The journey, for most customers, had ten steps, which included getting to the station, finding parking, buying tickets, locating the platform, and so on. The insight that proved most striking was that passengers did not take their seats on the train until stage eight—most of the experience of train travel, in other words, did not involve the train at all. The team reasoned that every one of the prior steps was an opportunity to create a positive interaction, opportunities that would have been overlooked if they had focused only on the design of the seats. Admittedly, this approach made the project far more complex, but that is typical in the move from design to design thinking. It may not be easy to reconcile the many interests that come into play in getting from Washington to New York, but Amtrak managed to do so and has created a more complete and satisfying experience for its customers. Despite its numerous and well-publicized problems with tracks, brake systems, and wheel sets, Acela has proved to be a popular service. The customer journey was the first prototype in that process.

acting out

If playing with Legos is a child’s way of “learning with your hands” and foam core and computer-driven milling machines are the equivalent for grown-up product designers, what does it look like for service innovation—the experience a person may have at a bank, a clinic, or the Department of Motor Vehicles? Our most reliable consultants, here as with so many other prod-
ucts, are kids. As soon as two or three children get together, they start to role-play: they become doctors and nurses, pirates, aliens, or Disney characters. Without prompting, they begin to perform lengthy enactments full of complex plots and subplots. Research suggests that this form of play is not only fun but also helps establish internal scripts by which we navigate as adults.

TownePlace Suites, an extended-stay hotel brand owned by Marriott, serves business travelers, such as consultants with long-term contracts, who may be required to be away from home for more than just a few nights and want to feel more at home than is usually the case in hotels. They are likely to work in their rooms more regularly, they stay over on weekends, and they may spend time on their own exploring the neighborhood. Marriott wanted to rethink the highly specific experience of these travelers.

Traditionally, one of the problems with architectural design is that full-scale prototyping is virtually impossible because it is just too expensive. Instead, an imaginative team of “space designers” rented an old warehouse in a dicey part of San Francisco’s Bayview district, where they built a full-scale mock-up of the entrance lobby and a typical guest suite of foam core. Their mock-up was not intended to showcase the aesthetic qualities of the space. Rather, it served as a stage on which designers, the client team, a group of hotel owner-operators, and even “customers” could act out different service experiences and explore in real space and real time what felt right. All the visitors were encouraged to add Post-its to the prototype and to suggest changes. This process yielded a host of innovations that included personalized guidebooks with local information tailored to repeat clients and their specific needs as well as a
huge wall map in the lobby where guests could use magnetic tiles to mark interesting restaurants or other landmarks—a sort of “open-source guestbook.” This full-scale space for acting out whatever occurred to them gave the design team a rich set of ideas for further testing. Moreover, they had a much better sense of how good the ideas were. No amount of survey work or virtual simulation would have achieved the same result.

Learning to feel comfortable acting out potential ideas is obviously important for anyone contemplating an experiential approach to prototyping—Mattel’s Ivy Ross went so far as to teach new recruits to the Platypus program how to use improvisational acting techniques in the first couple of weeks of the session. Knowing some of the basics, such as how to build on the ideas of one’s fellow actors and being willing to defer judgment of them, increases the likelihood that collaborative, real-time prototyping will be successful. The amateur theatrics of an experiential prototype can look foolish. It takes a certain confidence for individuals to loosen their ties, slip off their heels, and explore an idea through improvisation.

**prototyping in the wild**

Most prototyping takes place behind closed doors, for obvious reasons. It is often necessary to protect the confidentiality of ideas and limit their exposure so that the competition (and sometimes management) doesn’t know what’s up. Traditional companies may arrange focus groups or customer clinics, and edgier companies such as Electronic Arts regularly bring in gamers to test their games during development. Controlled
environments such as these work well enough in evaluating a product's functional characteristics: Does it work? Will it break when dropped? How well do the parts fit together? Will an average person be able to find the on/off switch? In fact, these are often aspects of a product that can be tested by the project team members themselves. Things become more complicated with services, however, and particularly with services that rely on complex social interactions. Mobile telephony, for example, draws on intangible interactions of users with one another and with the system itself. Today's complex ideas require prototypes to be released into the wild to see how they survive and adapt.

When the German mobile phone company T-Mobile began exploring ways of creating social groups via mobile phones, the company believed that networks of like-minded individuals could use phones not just to stay in touch but to share pictures and messages, make plans, synchronize schedules, and facilitate a hundred other interactions in a much more immediate way than with a PC. It would have been possible to create scenarios and storyboards to describe T-Mobile's ideas, and even to create simulations to run on phones. But the social dimension of the problem would have been overlooked. The only way to achieve this was to launch a prototype service. The design team loaded two prototypes onto some Nokia phones and handed them out to small groups of users in Slovakia and the Czech Republic. In less than two weeks it was clear which of the two prototypes was more compelling and why. The winning idea—helping users build social networks around events in their calendars—surprised the team, which had favored the alternative idea—helping people to create shared phone books. By launching prototypes, the team not only gathered real evidence of how
the new service might be used but avoided chasing after its own less promising idea. There was only one flaw in the innovative methodology: at the end of the trial, several of the users refused to give back their phones.

Another emerging form of “prototyping in the wild” involves the use of virtual worlds such as Second Life or social networks such as MySpace and Facebook. Companies can learn from consumers about proposed brands or services before they invest in the real thing. One successful example is the Starwood hotel chain, which launched a 3-D, computer-generated prototype of its planned Aloft brand inside the virtual world of Second Life in October 2006. Over the next nine months virtual guests inundated Starwood with suggestions on everything from the overall layout down to putting radios in the showers and repainting the lobby in earth tones. When enough feedback had been collected, Starwood shut down the virtual hotel to “renovate.” When it reopened, a gala cyberparty erupted in which hip avatars danced in the lobby, flirted in the bar, and hung out around the pool. And what do you do with an expensive virtual prototype once real construction begins? Starwood donated its abandoned “sim” to the online youth empowerment group TakingITGlobal.

Starwood’s Aloft brand wanted to capture a youthful, urban, stylish, and tech-savvy clientele—just the types likely to be found cruising the neighborhoods of Second Life. But the advantages of virtual prototyping make it likely that other, more conservative businesses will begin to experiment with it. Virtual prototyping allows companies to reach prospective customers quickly and get feedback from people in numerous locations. Iterations are easy, and as more of them begin to explore
the prototyping potential of online social networking, we will become increasingly adept at evaluating them. Like any prototyping medium, however, there are limitations. Virtual worlds such as Second Life rely upon avatars that represent customers, but we have no idea who they really are. This can be risky, as things are not always as they appear.

minding your own business

It is one thing to talk about prototyping material objects and even intangible services, but there is also a role for prototyping more abstract challenges, such as the design of new business strategies, new business offerings, and even new business organizations. Prototypes may bring an abstract idea to life in a way that a whole organization can understand and engage with.

HBO, famous for bringing us shows such as *The Sopranos* and *Sex and the City*, had by 2004 come to realize that the TV landscape was changing. It had earned its dominance in cable TV by delivering premium content, but the company could see that new delivery platforms such as Internet TV, mobile telephony, and video on demand were destined to become more important. HBO wanted to understand what the impact of these changes might be.

After a lengthy process of research and consumer observation, a strategy emerged based on creating seamless content that would spread across all of the emerging new technology platforms: desktop PCs, laptops, mobile phones, and Internet protocol television (IPTV). HBO, we concluded, should be willing to loosen its identification with cable TV and become
"technology agnostic," bringing content to customers whenever they wanted it and wherever they were. Instead of making a TV program and then thinking about what to do with DVDs or mobile content, shows should be created with these other channels in mind from the outset. We understood that this ambitious agenda challenged some fundamental premises. It required HBO not only to gain a deeper understanding of how audiences relate to media but also to break down some of the entrenched silos that existed within the company itself.

To create a compelling vision of the customer experience, the project team built prototypes and installed them in a walkthrough experience on the fifteenth floor of HBO’s New York headquarters. This enabled senior executives to see firsthand how customers might interact with TV content that they could access from different devices. For technical and analytical grounding, they constructed a future road map that ran the entire length of a wall and displayed the elements of technology, business, and culture that the company would confront as the program moved forward. Touring the fifteenth-floor environment we’d created, Eric Kessler, vice president for Marketing, got it: “This isn’t about the future of HBO On Demand. It’s about the future of HBO.”

The prototype projected HBO management into the future in a compelling, realistic way, helping them visualize both the opportunities and the challenges to come. When HBO entered into discussions with Cingular (which is now AT&T Wireless) to put premium TV content onto a mobile platform, the fifteenth-floor prototype helped them to reach a common understanding.
phase shift: prototyping an organization

HBO illustrates the need to think with our hands even when working at the level of business strategy, and the same is true for the design of organizations themselves. Institutions must evolve with changing environments. Though the company “re-org” has become a cliché in business culture, it is nevertheless one of the most fateful and complex design problems any company may face, though it is rarely accompanied by any of the basic characteristics of good design thinking. Meetings are called in which there is no brainstorming; organizational charts are drawn up with little evidence of any thinking with the hands; plans are made and directives are issued without the benefit of prototyping. I don’t know if IDEO could have saved the American auto industry, but we would have started with foam core and a hot glue gun.

To be sure, prototyping new organizational structures is difficult. By their nature, they are suspended in webs of interconnectedness. No unit can be tinkered with without affecting other parts of the organization. Prototyping with peoples’ lives is also a delicate proposition because there is, rightly, less tolerance for error. But despite this complexity, some institutions have taken a designer’s approach to organizational change.

The implosion of the dot-com supernova at the end of 2000 created a black hole whose epicenter was the San Francisco Bay Area. Designerly lofts were abandoned throughout San Francisco’s “Multimedia Gulch,” leaving only Aeron chairs and colorful iMacs; the $100,000-a-month billboards along Highway 101, the main corridor through Silicon Valley, fell empty; would-be entrepreneurs returned to college to finish
their degrees. IDEO, which had been working with new start-ups while helping more established companies navigate the passage into the Internet age, was hit hard. For the first time in our history, we experienced a forced belt-tightening. I had been summoned back from the United Kingdom, where I was heading up IDEO’s European operations to take over the reins of leadership from David Kelley, who, with his exquisite sense of timing, had decided to step down just minutes (or so it seemed) before the e-bubble burst, to focus on his academic life at Stanford. It fell to me to oversee the transition to IDEO 2.0.

From a company that had once boasted that it would never grow beyond forty employees (so that we could lock the front door, jump onto a school bus, and drive to the beach), we had now expanded nearly tenfold, and although we worked hard to preserve a flat organizational structure, that growth translated into 350 careers, benefits packages, and dreams to fulfill. The stakes were high and there was no safety net, so I decided to do what designers do: I put together a team, and we launched a project. The brief? To reinvent the firm.

Having spent the previous two decades creating a human-centered design process for our clients, it would have been odd indeed if we had not applied it to ourselves. That is precisely what we did. During “Phase One” the project team fanned out across the landscape, talking to designers in each of our offices, our clients, our network of collaborators, and even our competitors to gain insight into how the field was evolving, where we were weak, and where we were strong. These discussions led to a series of workshops and our first prototypes, which took the form of a cluster of “Big Ideas” that captured the future as we saw it. One of these was the idea of “design with a small
d”—using design as a tool to improve the quality of life at every level, as opposed to creating the signature objets that grace the pedestals of art museums and the covers of lifestyle magazines. Another was the idea we called “One IDEO,” the notion that our future depended on our acting not as independent studios but as a single interconnected network. A third idea was to abandon our original “studio” model—which reflected the way designers are organized—and replace it with a new, untested structure of “global practices” intended to reflect the way the world itself is organized: the “Health Practice” would focus on projects from precision medical equipment for Medtronic to educational packaging for GlaxoSmithKline; “Zero20” on the needs of kids from early infancy through late adolescence; other practices would be focused around interactive software, consumer experiences, the design of “smart spaces,” and even organizational transformation. At this point we felt that we were ready to take our prototypes out into the field. Or, to be more precise, we took the field to the prototypes.

We decided to stage a global event that, for the first time since we had expanded beyond our base in Silicon Valley, would bring together every employee of IDEO in one place: senior mechanical engineers from Boston, newly hired graphic designers from London, model makers from San Francisco, human factors specialists from Tokyo, and even our beloved receptionist Vicky in Palo Alto converged upon the Bay Area to jump-start what we soon began to call IDEO 2.0. Standing up in front of that audience of 350 peers, colleagues, and mentors to launch the event remains the high point of my career. Little did I know that the kickoff was the easy bit.

The launch—three days of lectures, seminars, workshops,
dancing, and a mass version of the old computer game Pong with 350 simultaneous players—was a huge success. The following year, however, was one of the toughest I have ever experienced. As the prototypes unfolded, we learned that a story needs to be repeated many times before people understand how it applies to them and many more times again before they change their behavior. We learned that leadership teams that had been successful with small local groups might not easily project their ideas across seven locations. We learned that visionary designers who had been accustomed to complete creative autonomy did not happily adapt to the idea of market-driven practices.

We redesigned IDEO because we wanted the organization to remain flexible, nimble, relevant, and responsive to the new global environment that was taking shape. Five years on, two of the original seven practices no longer exist, a new one has been added, and one has refashioned and renamed itself twice to find better resonance with its intended clients. When it comes to organizations, constant change is inevitable and everything is a prototype. At the most challenging times we reminded ourselves that a successful prototype is not one that works flawlessly; it is one that teaches us something—about our objectives, our process, and ourselves.

There are many approaches to prototyping, but they share a single, paradoxical feature: They slow us down to speed us up. By taking the time to prototype our ideas, we avoid costly mistakes such as becoming too complex too early and sticking with a weak idea for too long.
I wrote earlier that all design thinkers, whether or not they happen to have been trained in any of the recognized design disciplines, inhabit three “spaces of innovation.” Since design thinkers will continue to “think with their hands” throughout the life of a project—aiming toward greater fidelity as it advances toward completion—prototyping is one of the practices that enable them to occupy all three realms simultaneously.

Prototyping is always inspirational—not in the sense of a perfected artwork but just the opposite: because it inspires new ideas. Prototyping should start early in the life of a project, and we expect them to be numerous, quickly executed, and pretty ugly. Each one is intended to develop an idea “just enough” to allow the team to learn something and move on. At this relatively low level of resolution, it’s almost always best for the team members to make their own prototypes and not outsource them to others. Designers may require a fully equipped model shop, but design thinkers can “build” prototypes in the cafeteria, a boardroom, or a hotel suite.

One way to motivate early-stage prototyping is to set a goal: to have a prototype ready by the end of the first week or even the first day. Once tangible expressions begin to emerge, it becomes easy to try them out and elicit feedback internally from management and externally from potential customers. Indeed, one of the measures of an innovative organization is its average time to first prototype. In some organizations, this work can take months or even years—the automobile industry is a telling example. In the most creative organizations, it can happen within a few days.

In the ideation space we build prototypes to develop our
ideas to ensure that they incorporate the functional and emotional elements necessary to meet the demands of the market. As the project moves forward, the number of prototypes will go down while the resolution of each one goes up, but the purpose remains the same: to help refine an idea and improve it. If the precision required at this stage exceeds the capabilities of the team, it may be necessary to turn to outside experts—model makers, videographers, writers, or actors, as the case may be—for help.

In the third space of innovation we are concerned with *implementation*: communicating an idea with sufficient clarity to gain acceptance across the organization, proving it, and showing that it will work in its intended market. Here too, the habit of prototyping plays an essential role. At different stages the prototype may serve to validate a subassembly of a subassembly: the graphics on a screen, the armrest of a chair, or a detail in the interaction between a blood donor and a Red Cross volunteer. As the project nears completion, prototypes will likely be more complete. They will probably be expensive and complex and may be indistinguishable from the real thing. By this time you know you have a good idea; you just don’t yet know how good it is.

McDonald’s is a company famous for applying the prototyping process throughout each of the spaces of innovation. In the inspirational space, designers use sketches, quick mock-ups, and scenarios to explore new services, product offerings, and customer experiences. These might be kept under wraps or shown to management or consumers to get early feedback. To nurture the ideation space, McDonald’s has built a sophisticated prototyping facility at its headquarters outside Chicago where
project teams can configure every type of cooking equipment, point-of-sale technology, and restaurant layout to test new ideas. When a new idea is almost ready for *implementation*, it will often be tested in the form of a pilot deployed at selected restaurants.
Due Date: 10/16/2014

Book Title: Prototyping: a practitioner's guide /

Author: Warfel, Todd Zaki.

Library: 28654244

Reserve Users:

Duke Library

Send at:
delivery@duke.edu

Library

Duke Commons

t St. NW

DD

30332-0900

494-884-4511

delivery.gatech.edu
Every year, millions of people get a glimpse into the future—the concept car. Manufacturers invest several years and millions of dollars into these one-of-a-kind creations. Most of them never make it to mass production. Those that do are often a fraction of the original vision.

The automotive industry is highly competitive. Innovation is not only a means to stay ahead, but also often one of survival. Each concept is an exercise in design, an exploration into what’s possible, what’s feasible, and what’s marketable—it’s a prototype.

This method of prototyping has been a core part of the auto industry for decades. While these concept cars are expensive, it’s much more expensive to retool all the necessary machines and launch a failure. The risk is too high.

Making the argument that prototyping is a necessary part of the overall design process for something like a car or a missile guidance system is a no-brainer. However, in the world of software development, the argument for gaining buy-in for creating a prototype is a bit more challenging. In fact, it’s typically one of the greatest challenges we have.

In this first chapter, I’m going to highlight some of the challenges faced when trying to incorporate prototyping into an existing design/development process. I’m also going to give you a few pointers to show how valuable prototyping can be to identify problems early on, reduce risk, and ultimately save time, effort, and money.

Clients and management who aren’t familiar with prototyping often see it as a cost with little, if any, benefit. It’s one of the most common questions I’ve received. “How do I get my boss or client to buy into prototyping? They say we don’t have the time or budget for it.”

If your business is involved in building Web sites, software applications, or systems that have both a hardware and software component, you can’t afford not to prototype. As the complexity of the system increases, the cost-to-benefit ratio of prototyping increases dramatically.
Prototyping does have a cost. It isn’t free. But if you haven’t been prototyping, you’ve been missing opportunities for innovation and significant cost savings. The benefits of prototyping far outweigh the initial cost.

**Prototyping Is Generative**

One of the fundamental values of prototyping is that it’s generative, which means as you work through the prototyping process, you’re going to generate hundreds, if not thousands, of ideas. Some of them are brilliant and some are less brilliant. I’ve found that even those less brilliant ideas can be a catalyst for brilliant solutions.

As a generative process, prototyping often leads to innovation and a significant savings in time, effort, and cost. Prototyping helps you get ideas out of your head and into something more tangible—something you can feel, experience, work through, play with, and test.

**Prototyping—The Power of Show, Tell, and Experience**

If a picture is worth a thousand words, then a prototype is worth 10,000. Prototypes go beyond the power of show and tell—they let you experience the design.

“It’s one thing to talk about them and have storyboards and another thing to see them for real.”

—Robert Hoekman, Jr.

There are a number of ways to communicate or document a design, including requirements documents, wireframes, visual comps, and prototypes.

**Common Design Documentation Models**

**Requirements documents.** These are typically a written document describing the technical or functional requirements of a system.
Requirements documents tend to be more focused on written description and less on visual illustration—they are more *tell* and less *show*. The lack of visual simulation often leads to misinterpretation of a requirement. Screen shots can be included to help reduce this misinterpretation, but static screens only go so far.

**Wireframes.** Ever seen architectural blueprints for a house? Well, that's kind of what a wireframe is for software. Wireframes are a visual representation of the functional page structure. They visually communicate what functional pieces are present on a page and their relationship to each other. Wireframes are typically in black and white or shades of gray.

Combined with detailed behavior notes, wireframes do a better job at show and tell than requirements. However, wireframes often leave gaps in the design. These gaps result in missing details or misinterpretation, which is bad.

**Prototypes.** A prototype is a representative model or simulation of the final system. Unlike requirements documents and wireframes, prototypes go further than show and tell and actually let you experience the design.

Some technical requirements, like a 100KB page limit, might not be obvious in a prototype. These can easily be captured with a supplemental document much smaller than 60–200 pages.

On its own, a requirements document or wireframe is insufficient for show and tell of complex systems. You might be able to get by using one of these for simple systems, but for complex systems, you'll ultimately run into trouble. Oftentimes, they are used together in the attempt to create a "full picture." However, they still fall short when it comes to actually experiencing the design.

Combining annotated wireframes with a requirements document can get you to a 70–80 percent accuracy of the original vision. That's still too much room for error in my book.

**The AJAX and RIA Monkey Wrench**

Now, what happens if you throw AJAX or other Rich Internet Applications (RIAs) into the mix? Things start to fall apart—rapidly. Neither a
requirements document nor annotated wireframes do a good job at telling
the story of rich interactions and transitions.

Unlike traditional page-based interactions, AJAX and RIAs often leverage
state-based interactions. A page or screen can have several tiles or widgets
that operate independently and interdependently of each other. Updating
an RSS feed on a page doesn’t require refreshing the entire page anymore.
Instead, only the RSS feed widget updates, leaving the rest of the page alone.

This has prompted many in the design community to claim that the page
paradigm is dead—the new paradigm is the screen or state.

Transitions and animations are another challenge. Have you ever tried to
describe a self-healing AJAX transition? My best description, coupled with
some strategic hand waving and magic wand simulations, still results in
raised eyebrows and questionable looks.

As the presence of AJAX and other RIA technologies continue to grow,
the need for and value of prototypes as a design communication tool
dramatically increases. In fact, I would argue it becomes critical for success.

**Prototyping Reduces Misinterpretation**

Take a 60-page requirements document. Bring 15 people into a room.
Hand it out. Let them all read it. Now ask them what you’re building.
You’re going to get 15 different answers. Imagine trying the same thing with
a 200-page requirements document—it gets even worse.

Prototypes are a more concrete and tactile representation of the system
you’re building. They provide tangible experiences.

Once my company made a shift away from a requirements-dependent
process to a prototype-dependent process, we saw an immediate reduction
in the need for clarification and rework. We’ve gone from a 60–80 percent
consensus on interpretation to 90 percent or better.

We’ve also found that the total amount of time and effort required to
produce the prototype is less than that required to create a detailed
specification document and annotated wireframes.

**The Value of Prototyping**
I've found a number of reasons that written documentation leaves more room for misinterpretation:

- Nobody wants to read a 60–200-page written specification. There's really no joy in it.
- If you can't get them to read it, you won't get them to fully understand it.
- Written documentation doesn't allow you to see the "big picture." Instead, you're forced to see one line at a time.
- Words leave too much room for interpretation.

Prototypes, on the other hand, have a number of advantages that help reduce misinterpretation:

- You experience how the system would work, rather than just read about it.
- Prototypes encourage play. When you get someone to play with your prototype, you increase the likelihood that they'll understand it.

**Prototyping Saves Time, Effort, and Money**

How many times have you heard one of the following from a client, one of your bosses, or even a fellow designer or developer?

"We don't have time to prototype."

"We can't afford to prototype. We don't have the budget for it."

I've heard each of these arguments dozens of times. Frankly, they're not without some merit. As I said earlier, prototyping isn't free, but the benefits of prototyping far outweigh the cost of prototyping, or most importantly, not prototyping.

Talk to anyone who has made the transition from a design and development process that didn't include prototyping to one that does, and they'll tell you it has saved them a ton of time and headaches. Not only does prototyping let
you realize and experience the design faster, but ultimately it also reduces the amount of waste created by other design and development processes.

**Prototyping Reduces Waste**

In a typical design and development process, requirements are written and handed off to a designer or developer. The designer or developer then interprets these requirements and builds something based on his/her interpretation.

Theoretically, a requirements-driven design process should reduce waste. The overall goal is to get everyone on the same page. If we’re all on the same page, ultimately, we’ll have less waste. Sounds fantastic.

Theoretically, it’s a very sound idea. As experience will show, however, theory and reality are often very different. There are a number of shortcomings in a traditional requirements-driven design and development process that create waste, and they include the following:

1. **Written by the wrong person.** Designers and developers are rarely included in the requirements writing process. Instead, the requirements are often written by a business analyst or his equivalent. This person lacks the technical and design knowledge of their counterparts, which often results in any number of requirements being rewritten several times.

2. **Significant time and effort.** The amount of time invested in writing, reviewing, and revising these detailed requirements is significant. For complex systems, I’ve seen it take 3–9 months to finish something—sometimes more. During that time, things change.

3. **Non-final final.** Theoretically, the requirements are the final documentation. In reality, requirements are constantly changing, even after they’re “complete.”

4. **Misinterpretation.** The amount of misinterpretation of the 60–200-page requirements is often significant. Misinterpretation leads to weeks or months of rework and a delayed product launch.
5. **Nonessential features.** Requirements are often filled with features that provide little, if any, value. Those features take time and effort to build and test. This results in wasted time in writing requirements, building, and testing features that provide little, if any, value and often go unused.

6. **Catching mistakes too late.** Requirements-driven processes typically won’t catch a mistake until it’s in production. The later you catch a mistake in the development process, the more costly it is to fix.

Any one of these items alone creates wasted time and effort. Typically, a requirements-driven process is plagued with several of these issues, creating a great deal of inefficiency and waste. On the other hand, including prototyping in the process can help reduce the amount of waste and result in these benefits:

1. **Decisions by the right people.** Designers and developers can flex their experience and knowledge, contribute to the process, and ultimately ensure that the right people make the design decisions.

2. **Survival of the fittest.** Multiple ideas are created and tested to ensure that the strongest solutions survive.

3. **Adaptive.** Prototypes can be quickly updated, compensating for the ever-changing nature of software development.

4. **Reduced misinterpretation.** The prototype is a visual, or sometimes physical, representation of the system. Visual and physical representations leave less room for misinterpretation than a 60–200-page written document. By reducing misinterpretation, you reduce the amount of rework. Less rework means lower costs and faster time to market.

5. **Focus.** Prototyping produces more focused products. More focused products produce less waste in design, development, and rework.

6. **Catch mistakes early.** Prototyping helps you catch mistakes early in the design and development process. The earlier you catch a mistake, the lower the cost to fix it will be.
7. **Reduce risk.** Prototyping reduces risk, by reducing misinterpretation and catching problems earlier in the design and development cycle.

While prototyping can’t solve all the problems that plague requirements-driven processes, it can definitely help reduce many of the more common inefficiencies and waste.

**Prototyping Provides Real-World Value**

Jonathan Baker-Bates is someone who has seen a measurable benefit from prototyping firsthand. Jonathan works for a consulting company in the UK with a very typical design and development story. His team of developers regularly receives a 200-page specification document to quote against and build to. Well, that was what they used to do.

Jonathan’s company recently made a shift toward a prototyping-oriented process. Instead of giving developers a 200-page document, they now receive a high-fidelity prototype with a 16-page supporting document.

Since the change, his company has noticed a number of significant improvements:

- Time and effort required to produce the prototype and 16-page supplemental document is less than half required for the 200-page specification document.

- Estimates for build time and cost have become 50 percent more accurate.

- Request for clarification by the development team has been reduced by 80 percent.

- The amount of rework and bug fixes post-launch has been reduced to 25 percent of similar previous projects.

- All team members agree that executing the design with the prototyping process is easier than the old process.
Prototyping is commonplace in other design fields like architecture and industrial design. In fact, it’s not just accepted, but expected.

Why isn’t it as expected in software development? After all, software development, architecture, and industrial design have so much in common, including the following characteristics:

- They are all design processes.
- Artifacts are produced to communicate the design.
- The end result is a tangible product that people can experience and use.

I think the first reason is that in software development, the emphasis is often placed on the development process and not the design process. The industry doesn’t call it “software design”; they call it “software development.”

In software development, design is often an afterthought. The emphasis is on the technology or features—not the design. In architecture and industrial design, however, the emphasis is on design. Form follows function.

Another reason is that software development is seen as a manufacturing process, but architecture and industrial design are seen as a craft.

Perhaps these shortcomings are the result of how each field is taught. Computer science classes focus on teaching students technologies. Architecture and industrial design, on the other hand, focus on teaching students design principles and include something called design studio.

**The Absence of Design Studio**

In the world of architecture and industrial design, a design studio is a process, not just a physical place. This process is taught in every respectable architecture and industrial design program. You’ll be hard pressed to find a design studio class in computer science.
In studio classes, you design or prototype and present to your peers. Your peers critique your work, highlighting the strengths and areas that still need some work.

When you design, present, and critique, think of it as a collaborative, rapid, iterative design process. During this process, you get to share in the ideas, successes, and failures of your peers.

Design studio is a core part of architecture and industrial design. Prototyping is a core part of that studio process. So students learn this skill early on, and they use it regularly. Prototyping becomes a regular tool in their design process.

What Does the Prototyping Process Look Like?

First, let’s be clear about something—there is no uber-prototyping process, a single formula like there is for Coca-Cola. However, there are a number of tried-and-true guiding principles you can follow, no matter what you’re prototyping.

Second, no matter what prototyping process you decide to use, keep in mind that a process is merely a means to an end. One of the most common pitfalls of following a process is being tied too tightly to it. If you’re more focused on the process than the end goal, you won’t be successful.

Third, a good process balances structure with flexibility. It provides a solid foundation, while giving you the flexibility to change and adapt that process as needed, due to changes in time or circumstance.

Finally, a good process breeds success. When your process starts to limit success, then you need to revisit it.

Mark Sanders, the inventor of the Strida folding bicycles, describes the process he used to create the bikes on YouTube\(^1\). As I watched Mark describe his process, a few things stood out:

- He didn’t worry about which ideas were good and which were bad. Instead, he used sketching to explore every possible idea.

\(^1\) toddwarfel.com/archives/great-video-on-prototyping-bikes/
Once he had plenty of sketches, he evaluated them based on the original goals of the product. This left him with the best ideas.

Sketches only go so far. In order to see which of his best ideas would really work, he had to build models or prototypes.

Sketching also played a critical part in updating the design to create the Stridaz.

The process Mark describes in the video is a pretty typical prototyping process, which includes the following elements:

- Sketching
- Evaluation
- Modeling
- Testing

At my company, Messagefirst, we use a similar approach to the one Mark describes, but with a small twist. Our prototyping process takes a page from the design studio book and uses the presentation and critique model for evaluation. So our modified approach looks like this:

- Sketching (e.g., whiteboards, paper, code)
- Presentation and critique
- Modeling (prototyping)
- Testing

Since one of the goals of our process is rapid iteration and evolution, we stress some pretty short timeframes on the first two stages—sketching and presentation and critique. This emphasis on short cycles keeps the process moving and makes us more productive. It also enhances the generative nature of prototyping.

Our prototyping process is also iterative and evolutionary. We sketch, present and critique, prototype, present and critique, sketch, present and critique, prototype, present and critique, prototype, and test (see Figure 2.1). Then we do it all again.
One thing you might have noticed is the cyclical nature and multiple instances of the first two steps: sketching and presentation and critique. Sketching is woven throughout our prototyping process. Anytime we sketch, we present and critique. In fact, when we’re presenting and critiquing our prototypes, either internally or externally with a client, we sketch our revisions.

Now, let’s taking a detailed look at the prototyping process I use, starting with sketching.

**Part 1: Sketching**

Sketching is the generative part of prototyping. As part of the generative nature of prototyping, your goal is to get the ideas out of your head and into a more tangible format.

I prefer putting a time limit on “sketching time.” This forces the sketchers to work quickly without getting caught up in the details.

The goal of sketching isn’t to flesh out your ideas fully—you’ll do that during the prototyping stage of the process. The goal is to generate a number of concepts, get them out of your head as quickly as possible, and move on.
One of the benefits of sketching with code is the inherent ease of turning your sketches into a prototype. With the increasing number of JavaScript libraries, CSS frameworks, and application frameworks like Ruby on Rails, sketching in code is easier than ever.

**Advantages**

- It makes sketching in code easier with an increasing number of available tools.
- It brings your sketches to life—you can actually play with your sketches.
- It leverages existing code, when possible.

**Disadvantages**

- Not everyone can code.
- It requires a computer.
- It's less collaborative than paper or a whiteboard.
- It takes more time than paper or a whiteboard.

**Sketching on Whiteboards**

One of the biggest benefits of sketching on whiteboards is its inherent collaborative nature. It's very easy for anyone to participate in the discussion and sketch on the whiteboard.

**Advantages**

- It's collaborative.
- Anyone can draw on a whiteboard.
- More than one person can participate at a time.
- No computer is necessary.
- Revisions are easy—just erase and draw again.
Disadvantages

- It’s less portable than code or paper.
- Sketches are still static.
- Capturing sketches from a whiteboard can be difficult.

Sketching on Paper

Sketching on paper is still my favorite. Similar to the collaborative nature of sketching on whiteboards, paper is ultra portable and can be done anywhere.

Advantages

- It’s collaborative.
- Anyone can draw on paper.
- More than one person can participate at a time.
- No computer is necessary.
- Revisions are easy—just draw over your current sketch or grab another piece of paper.
- It can happen anytime, anywhere.
- It’s definitely portable.

Disadvantages

- Sketches are still static.

Part 2: Presentation and Critique

Presentation and critique is arguably the most important part of our prototyping process. This is where we focus on quality.

Presentation and critique is a technique I learned from studio class during my undergraduate days while studying graphic design. And even though

---

3 Sketches can be captured with photos or using a Smartboard.
I later changed my degree to English and Cognitive Psychology, I'll never forget the very valuable lessons learned from presentation and critique.

The goal of the presentation and critique stage is to find the best ideas. You present the strengths of your concept, and your peers highlight areas that need work or further clarification. That's it—discuss, evaluate, and move on.

When presenting our sketches for critique, we often taped them up on a wall, as shown in Figure 2.4.

![Presentation of sketches during a design studio critique session.](image)

**FIGURE 2.4**
Presentation of sketches during a design studio critique session.

**Presentation and Critique Guidelines**

**Keep it short.** As I mentioned earlier, presentation and critique is the second stage where we emphasize tight timelines. In fact, presentation and critique time is shorter than sketching time. I like to limit presentation time to two minutes and critique to three minutes.
It's important not to spend too much time during the presentation and critique step. I know, it sounds a bit counterintuitive. But just like when you're sketching, the purpose is to get something out quickly and move on. You're going to refine your ideas later.

**Three-minute presentation per concept.** A time limit of three minutes per presentation per concept means that you'll have to focus on the strongest parts. And if you can't explain your concept in less than two minutes, then something is probably wrong with it.

**Two-minute critique.** Your peers get two minutes to critique your concept. During the critique, they have to provide two to three things they think are strong about it and one to two areas that need improvement or need to be worked out a bit more.

**Take notes.** Write directly on your prototype sketches. It's okay. It's just paper. Use the critique of your peers to refine and strengthen your design concept.

**Part 3: Prototype**

At this point in the process, you've sketched out your ideas, presented and had them critiqued, and are now left with the strongest concepts. These are the ones you're going to prototype.

Prototyping is where you start to work out the details of your design and figure out which ones will actually work.

When I prototype, I consider the following details:

- Am I using a tool or medium I'm comfortable in?
- Do I have the ability to communicate effectively what I need to the *audience or consumer*?
- How much time do I have?
- What level of fidelity do I need?
It really doesn’t matter what you build your prototype in at this stage. Most of my prototypes are HTML/AJAX prototypes, but that’s due to the nature of my work and clients. I’ve also created Flash, Keynote, and paper prototypes.

Once I have a prototype, I repeat the presentation and critique stage. I use the same basic model, presenting a piece of the prototype at a time and offering it up for critique from the client or customer. The biggest difference is when I present a prototype, I increase the time limits on presentation and critique. Other than that, the process follows the same guidelines.

**TIP** PROJECT AND SKETCH

Project your prototype on a whiteboard during the presentation and critique process. This allows you to sketch on top of your prototype during the process.

**Part 4: Test**

I divide testing into one of two categories: testing with the client or testing with the end consumer.

*Testing with Clients*

Testing with the client is a hybrid model of presentation and critique with sketching. These sessions typically last 1.5–3 hours, depending on the complexity of the prototype.

During these sessions, I follow the basic presentation and critique model, presenting one piece at a time and opening it up for discussion. Rather than writing up a list of revisions, I use a sketchboard to create sketch notes. These sketch notes are mostly sketches with a few labels or handwritten notes added.

I’ve found that sketching out the revisions ensures that we all walk away from the table on the same page. Written notes leave too much room for misinterpretation. By sketching the revisions, the risk of misinterpretation is reduced. And since sketching is collaborative, the client can contribute easily to the solution.

Basically, I follow the prototyping process for defining revisions during the prototype process—more sketching, less writing.
Once the design review session is finished, the client gets access to a copy of the prototype, and I ask him to play with it for the next two to three days.

Since one of the goals of a prototype is to get it into the audience’s hands, I want the client to experience it, play with it, and use it. After he does, he’ll either find additional issues, or realize that something he thought was an issue really isn’t.

**Testing with End Customers**

Testing with the end customer is a standard usability test—8–12 participants, 5–6 scenarios, audio-video capture, analysis, and reporting of results afterward. Check out Chapter 12, “Testing Your Prototype,” for more detailed information on testing your prototype.

In either case, client or customer testing, I incorporate the feedback into the next iteration of the prototype.
Summary

So that's an overview of the prototyping process I use at Messagefirst. Hopefully, this has helped newbies see how prototyping can improve their design process. And for those seasoned pros, I hope you found a few tips to improve your current prototyping process.

Prototyping is a process, not a product—like any other process, it's only a means to an end. A few important points to remember when you're prototyping:

- Sketching is a key part of the prototyping process.
- Use the design studio method of sketching, presenting, and critiquing as a way to iterate rapidly.
- Start with quantity, exploring lots of ideas. Quality will come later.

In the next two chapters, we'll look at the five different types of prototypes and the eight guiding principles for better prototyping.
CHAPTER 3

Five Types of Prototypes

Type 1: Shared Communication 28
Type 2: Working Through a Design 33
Type 3: Selling Your Idea Internally 33
Type 4: Usability Testing 35
Type 5: Gauging Technical Feasibility and Value 37
Summary 42
Summary

While prototypes can vary in fidelity and functionality, the most common uses for prototypes include:

- Creating a shared communication.
- Working through a design.
- Selling an idea to your boss or team members.
- Usability testing.
- Gauging technical feasibility and value.

You may find your prototypes fit more than one of these uses. In fact, that's pretty common. It's fairly easy to see how a prototype could serve as a shared communication platform, a vehicle for usability testing, and a way to test technical feasibility.

In the next chapter we'll look at eight guiding principles for better prototyping.
CHAPTER 4

Eight Guiding Principles

Principle 1: Understand Your Audience and Intent 44
Principle 2: Plan a Little—Prototype the Rest 45
Principle 3: Set Expectations 46
Principle 4: You Can Sketch 48
Principle 5: It’s a Prototype—Not the Mona Lisa 50
Principle 6: If You Can’t Make It, Fake It 51
Principle 7: Prototype Only What You Need 53
Principle 8: Reduce Risk—Prototype Early and Often 53
Summary 56
Prototyping isn't as hard as you think. In fact, it's pretty easy. Anyone can prototype. Just like anything else, the more you do it, the easier it gets. But here's the catch—it's just as easy to mess it up.

Most of the mistakes I've made, seen, or heard about didn't happen from selecting the wrong tool or method. Instead, most of the mistakes came from the following situations:

- Prototyping either too much or too little.
- Prototyping the wrong thing.
- Not setting expectations for what the prototype will be.

Effective prototyping is about finding balance and setting expectations. In this chapter, I'm going to reveal eight guiding principles I've developed for more effective prototyping. These principles apply, regardless of method or tool.

Best of all, whether you're a seasoned protyper or just getting your feet wet, you'll benefit from the following eight guiding principles.

**Principle 1: Understand Your Audience and Intent**

This is the first and by far the most critical principle in the prototyping process. Understanding your audience for and the intent of the prototype drives every other aspect of the prototyping process. After you understand your audience and intent, you will be better equipped to:

- Determine what you need to prototype.
- Set appropriate expectations.
- Determine the right level of fidelity.
- Pick the right tool for the job.

Let's begin with addressing the question of audience, since it all starts here. When you understand who the audience is, you can determine what you need to prototype, how much you need to prototype, and what fidelity is appropriate for them.
If the audience is myself, another designer, or even an engineer, then a lo-fi paper prototype or a quick-and-dirty PowerPoint or HTML simulation is probably good enough. Those are mediums you can work with, you can understand, and that will get your point across without too much work.

However, if the audience for your prototype is a customer or a senior executive, chances are you’ll need something more polished. A cocktail napkin sketch probably won’t cut it.

When considering your audience, you should consider what medium or level of fidelity they will be comfortable with. If they can work with a few rough sketches on paper and you’re confident that is all you need to communicate your concept to them, then go for it. If, on the other hand, your audience is going to struggle with that medium and you’re going to struggle using it to communicate your concept to them, then pick a different medium or fidelity.

Once you understand your audience and the intent of your objective, you’re ready to begin your planning phase and start prototyping.

**Principle 2: Plan a Little—Prototype the Rest**

Software systems change constantly and quickly. By planning a little and prototyping the rest, you work incrementally and iteratively, making up for the ever-changing environment.

The more work you do in the planning process, the better off you’ll be. Of course, there is a point of diminishing return, so use some common sense.

I’m often asked how much planning should be done before you start prototyping. While there’s no magic number, I plan up to approximately 70 percent of the design through sketching and then it’s down to the business of prototyping.

Why 70 percent? There are two main reasons. First, since my goal is to get audience feedback, the faster I get it into their hands, the faster I can get feedback. Second, prototyping is a great tool for working through a design.
If I can get 70 percent of my design concept down on paper, then I can use prototyping to work through the rest.

In some ways, this makes prototyping a leap of faith. For those who are used to a waterfall method, or an environment where everything is “planned to a T” before you begin, this will probably make you a little uncomfortable. But just try it. Try planning about 70 percent of your prototype on a whiteboard or with paper and pencil and then starting prototyping. I’m confident you’ll like the results.

Naturally, each prototype is created on a case-by-case basis. Sometimes, you might need to plan a little more, or you can plan a little less. Mission-critical systems, like a missile defense system or a system for monitoring a patient’s vital signs in a hospital, will probably take a little more planning than say a video player.

There are other factors you might need to consider, such as environment, the tools you’re using, and so on. Your magic number might not be 70 percent. Experiment a little with the amount of planning-to-prototyping you do. You’ll find your sweet spot after a few tries.

When you plan a little and prototype the rest, you’ll see the system come together quickly. You’ll find your mistakes fast—and yes, there will be mistakes. You’ll be able to fix those mistakes with less total time and effort. Best of all, you’ll have something your audience can play with and ultimately give you feedback on that you can use in a timely manner.

**Principle 3: Set Expectations**

Setting expectations is based on a psychological technique known as priming. When you prime your audience, you guide their attention and focus.

Let’s try a little experiment. I’ve prototyped a shopping cart and checkout experience for the ecommerce area of a mobile service provider. In just a minute, I’m going to show you a prototype of some of the concepts which will highlight a few key features, such as featured products during the shopping experience and promoting phone accessories during the checkout process. Both of these key features will help increase profitability through add-on sales.
See how this works? I haven’t even shown you a prototype yet, but you have an expectation of what you’re going to see. You can even start to imagine them.

When I show you the prototype, you’re more likely to focus on looking for the featured product and promoted accessories than you are on what’s in the header and footer or what color the checkout button is.

Setting expectations typically boils down to two things: fidelity and functionality. For companies who are just starting to prototype, this is one of the most common mistakes and the easiest to avoid.

Remember the first guiding principle—know your audience and intent. If this is the first experience your audience has with prototyping, it’s critical you set their expectations of what to expect from the prototype. Reactions to a prototype are more favorable if the audience can predict with some degree of certainty what they will and won’t see.

By setting the expectation up-front, you avoid the rabbit-hole discussions about detailed interactions or pieces of functionality that simply haven’t been prototyped yet. Not to say that they won’t come up, because they will. By setting expectations correctly in the beginning, you give yourself an easy out—it’s not part of the prototype yet, but you can work it into the next release.

After you’ve primed your audience and set their expectations, launch the prototype and show them the demo. Don’t be afraid to discuss things that aren’t in the prototype at this stage, but try to keep the discussion focused on what is in this particular prototype. Remind them that this is a prototype and that some things might not be fully fleshed out yet.

**TIP CREATE A PLAYLIST SCRIPT**

Write a short “playlist” of the most important features that you want to highlight when demo’ing your prototype. It will help keep you on track and ensure that you don’t miss anything during the presentation.
Principle 4: You Can Sketch

We're so lo-fi it's not even funny.

—Scott Matthews
Xplane

At a small conference named Overlap, David Gray of Xplane asked by a show of hands who could draw. Of the 40 plus people attending only a few raised their hands. Then Dave asked another question, "Who here could draw when they were a kid?" Everyone in the room raised their hands. His response was simple, "So, what happened between then and now that you lost your ability to draw?" Nobody really had a good response.

Dave, with some help from his colleague Scott Matthews, went on to show how you could draw anything you need with a few simple shapes—a square, a triangle, a circle, and a straight line. He showed how you could draw a person running similar to the sketch in Figure 4.1.

FIGURE 4.1
Sketch of a running man.
You probably won’t find that particular drawing in the next issue of any anatomy book, but we all get the picture. And that’s what prototypes are—showing, communicating, and helping your audience get the picture.

I don’t consider myself to be an artist, but I do sketch—a lot. Some of my sketches are more refined than others. Sometimes I’ll put actual words for field labels, and other times I’ll just draw lines to represent that a label needs to be there, as shown in Figure 4.2.

If I’m doing an ultra quick-and-dirty sketch, where I’m really only interested in carving out parts of the screen real estate for functionality, I’ll go lower fidelity and typically just use lines. Or if I’m live sketching with another designer or the client, I’ll use the same technique.

If the actual order of the fields is critical and I need to communicate that, I’ll go a little higher fidelity and either write the labels in or possibly open Illustrator and use it to sketch out the screen.

This decision often comes back to the first principle: Know your audience and intent. If it’s just me, I’m often fine with lines and boxes, no labels needed. (I’ll have a list of those written on a separate piece of paper.) If it’s someone else who’s consuming it, I’ll usually put in a little more effort and write out the labels.
Remember if you could draw when you were a kid, you can draw now. Your goal isn't to create an illustration for the New Yorker, it's to communicate your idea. After all, it's just a prototype. Which brings us to Principle 5.

**TIP** **WHITEBOARDING AJAX**

Try using a whiteboard for sketching. You can simulate interactions and AJAX transitions through erasing and redrawing.

---

**Principle 5: It's a Prototype—Not the Mona Lisa**

Prototypes by their very nature are somewhat incomplete, sketchy versions of the final product. They're not perfect. They don't have to be. They're not meant to be. In fact, a slightly rough and sketchy prototype is often better for getting feedback.

If it's unfinished, participants are more apt to give their feedback. They don't feel that all the decisions have been made and are subsequently set in stone.

Admittedly, there are cases when you need something more refined. A sketchy prototype at a trade show probably isn't going to cut it. And your CEO might have some trouble envisioning the final product from a sketch or a black-and-white version of the prototype. So, again, use some common sense judgment.

What I can tell you with great confidence, however, is this—in most cases, your prototype doesn't have to be a Mona Lisa—good enough is good enough.

You're not shooting for perfection here—it's a prototype. You're shooting for something that takes the least amount of time and effort required to communicate to your audience the core concept of your idea. All you need is the right level of fidelity. No more. No less.
Principle 6: If You Can’t Make It, Fake It

This is probably the biggest hurdle for newbies venturing into prototyping. Whenever I give a talk or workshop on prototyping, I start by asking the audience a few questions:

- How many people feel comfortable writing code (either HTML or something else)?
- How many people have prototyped in some way, shape, or form in the past (either PowerPoint, HTML, Dreamweaver, PDFs, etc.)?

In general, I find that the more designers there are in the room, the fewer people I find that feel comfortable writing code of some kind and the fewer people who have prototyped or feel comfortable prototyping. This typically comes down to the myth that if you can’t write code, you can’t prototype. And this has increased with RIAs. If you can’t write JavaScript, then you can’t prototype.

As prototyping gets quicker and easier, it really does save a lot of time to quickly prototype the ideas people have and show them in use. It fleshes out the wheat from the chaff very effectively.

—Baruch Sachs

Director, Human Factors Design and Pegasystems

If you can’t code, or can’t make it, there are a number of options for faking it.

- You can fake it with a series of JPEG screens. Use Dreamweaver to create image maps and link them together. You don’t write a single line of code, but you can get feedback on the interaction and flow to see if it makes sense.
- Use Fireworks’ built-in capabilities to link between pages and frames; then generate a clickable HTML prototype.
- Use your favorite PDF creation tool or Adobe Acrobat and link them together for the same result.
- Use PowerPoint to link a series of still screens together.

- Use a series of HTML screens to simulate AJAX and other rich interactions.

At a talk I did for DC Refresh, I discussed a recent prototype we had done for a client. I talked about some of the rich interactions we had incorporated into the prototype with the help of the popular Prototype JavaScript library.

I told the audience I would show them a few basic show/hide interactions we had built into the prototype. I also told them that some of the interactions they would see were real and some were faked. And then I showed them the show/hide functionality for advanced search in Figures 4.3 and 4.4.

![FIGURE 4.3](image)
Basic search (hide).

![FIGURE 4.4](image)
Advanced search (show).

I toggled back and forth several times between the two states—showing advanced search and then hiding it. Then I pointed out the two different URLs. This AJAX simulation wasn’t AJAX at all, but rather two different screens linked together. It was faked.

The thing is, it didn’t matter that it was faked. The audience got the concept of show/hide for advanced search.

There are a number of tools out there to fake it, and more than likely you have more than one of them at your disposal. As long as you prime your audience first, set their expectations for what they’re going to see, and then have a simulation that shows what you’ve described, you’re good to go.
Principle 7: Prototype Only
What You Need

More often than not, the prototypes you build are going to be pieces of the entire system. You don’t need to build the entire system and get it to work to explore a design or to get feedback on it. In fact, trying to build the entire system loses the inherent benefits of rapid iteration.

If your ultimate goal is to use the prototypes for testing, chances are you’re going to test five to six scenarios. In that case, you only need to build what’s needed for those five to six scenarios.

What happens if a test participant clicks on part of the prototype that hasn’t been built? It’s a prototype. Prototypes are inherently incomplete. If a participant clicks on a feature that isn’t built, you use that as an opportunity to explore how he might expect it to behave.

By prototyping only the pieces you need, your investment is greatly reduced in a number of ways—cost, time, and effort. Additionally, since it takes less time to build only what you need, you’ll get feedback faster and move on. If the small piece you build works, then you can keep going. If not, then there’s no major loss, and you can try something else.

Principle 8: Reduce Risk—Prototype Early and Often

_We have these huge firms cranking out wireframes, building things, and discovering problems way too late._

—Anders Ramsay

As we’ve already discussed, prototyping has a number of advantages, one of which is the low investment-to-benefit ratio. Let’s look at two development models—one is the traditional waterfall method and the other utilizes rapid iterative prototyping.
In a traditional waterfall method, all of the system’s features and functions are planned before any development begins. This often leads to a six- to nine-month planning cycle before any actual work on the system begins.

In environments that move slowly or don’t change much, this might not be an issue. In today’s software industry, however, nine months is a lifetime—an entire company can be created bought, sold, and go under in nine months.

But let’s pretend your entire industry moves at the same slow snail’s pace that you do. In that case, you’re so heavily invested at this point that it’s nearly impossible or just not feasible to change direction. The ship has already sailed, and there’s no changing course—there’s no turning back.

This is a very costly model. More often than not, mistakes are made, and the recovery is too costly, as high as 100 percent. As such, these mistakes are left in the system for the end user to try to work around.

The other approach takes a more agile approach. You chip away at bite-sized pieces and use an incremental, iterative, and evolutionary approach. By using this approach with prototyping, you’ve only invested in small amounts. Reducing your investment reduces your risk.

You’re committing a few weeks at a time to a small set of concepts to see if they’re going to work or not. If they don’t work, there’s much less damage that’s done, because you’re only a few weeks away from pulling them and replacing them with something that works better. You’re not stuck for another six, nine, twelve, or eighteen months.

If they do work, you see an immediate benefit. You can make rapid iterative adjustments. You can stay ahead of your market, often leading the space. You can be more proactive and less reactive.

This is an area where prototyping really shines. You make small investments with a significant return. That return can be positive or negative. If it’s positive, then all the better. If it’s negative, then your risk is substantially reduced because you’ll find it early enough in the process and be able to replace it quickly.
The earlier in the development process you catch a mistake, the easier and less costly it is to correct your mistake. And don’t kid yourself—you will make mistakes. By some estimates, changes during design can be as low as 10 percent, and during development or after product launch, your costs skyrocket to 100 percent.

When you start prototyping early and often, you’ll reduce your risk and save yourself a great deal of headache, time, effort, and money.
Options for Using Omeka

Creating Your Own Omeka Installation

For the technically adventurous: If you want to set up your own instance of Omeka you need access to a web server. This can be done through a commercial hosting company or your institution might provide server space (either free or for a usually reasonable fee).

The server needs to offer the so-called LAMP software stack. LAMP refers to four software applications:

- **L** as in Linux: Linux is the operating system on which many servers run. Unlike more common operating systems, such as Windows or Mac OS, Linux is free, open-source software and comes in a wide variety of flavors.
- **A** as in Apache: Apache is a very popular HTTP server which means that it is the software that handles requests for web pages and other content that come in from external computers.
- **M** as in MySQL: MySQL is a database application, similar to applications like Microsoft Access. Most of the data you upload to Omeka will be stored in a MySQL database.
- **P** as in PHP: PHP is a scripting language; code written in PHP handles any interactions with the database and file uploads and generates the pages that you see.

If you want to run Omeka on your Windows or Mac computer, you can substitute a LAMP environment with XAMPP (Windows) or MAMP (Mac or Windows) that let you install Apache, MySQL, and PHP with a couple of mouse clicks.

For a good overview of the Omeka installation process, check out these two tutorials:

- From the developers of Omeka: https://omeka.org/codex/Installation

Using Omeka.net

Omeka.net is similar to what wordpress.com is for Wordpress blogs. It is a hosted service that takes all the technical work out of your hands; all you have to do is to sign up for an account at:

http://www.omeka.net/
Omeka.net offers a free as well as several paid plans. Plans vary based on the number of sites, plugins and themes and also on the amount of storage that will be available to you. The free basic plan gives you 1 site, 15 plugins, 5 themes and 500MB of storage which gives you a good starting point. If you want to learn more about Omeka.net, check out this link, http://info.omeka.net/.

Using the Omeka Installation for this Class

We have set up an Omeka installation that you can use for this workshop. You can find it at:

http://omeka.trinity.duke.edu

This site is mostly for getting experience with some of the more advanced Omeka plugins, like Neatline, you can also chose to use it for your own digital project for class. At the beginning of class, you will be sent an email asking you to create a password for the site. To login go to:

http://omeka.trinity.duke.edu/admin/users/login

This installation gives you a lot more plugins than Omeka.net, including the aforementioned Neatline plugin suite. However, there are a few drawback. Since the current version of Omeka lets you only run one site per installation, you would have to share one Omeka site with all other users of that site. Moreover, unlike an Omeka.net instance, the work you do on this site will not be saved for long after the class as it was set up only for this workshop.

Running a Virtual Machine with Omeka on Your Computer

Another option is to run a so-called virtual machine (VM) on your computer. A virtual machine is like having a separate computer inside your machine. It can run a totally different operating system and if something goes wrong, you can just delete it and create a new one. This virtual machine uses an operating system called Lubuntu; it is a lightweight (i.e., less resource-demanding) version of the popular Linux-based Ubuntu operating system. This virtual machine already has all available plugins and themes installed and gives you complete freedom to modify Omeka.

In order to be able to use a virtual machine, you need to first download and install a free software called Virtualbox. You can find versions for Windows, Mac OS and Linux at:

https://www.virtualbox.org/wiki/Downloads

Also, you will need to download a .vdi file. When downloaded on your computer, this file serves the same function as the hard drive of a physical computer and contains all the files and
software that your virtual machine will need. As the file is rather large (about 5.5GB), you should download the file before coming to DHSI; we will send you the link ahead of the Institute. We will go over the last steps of setting up the VM on your computer at the beginning of the workshop.

Since using the virtual machine means that your normal (physical) computer will be sharing resources—such as memory and processing power—with a virtual computer, your computer should fulfill certain minimum requirements in order to properly run the VM. It should have at least 4GB of memory and a multicore processor. If you are unsure of whether your computer meets these requirements, just contact us and we will help you figure it out.

To log into the VM use the password "omeka".

To log into the Omeka installation, use the following username and password:
username: dhsioomeka
password: dhsioomeka
Omeka/Neatline Tutorial

The following Omeka/Neatline Tutorial comes courtesy of Ronda Grizzle of the Scholar's Lab, University of Virginia.
Introduction to Omeka + Neatline

What is Omeka?
Omeka is an open-source, web-based exhibit builder developed by the Roy Rosensweig Center for History and New Media at George Mason University. Their goal was to create a universal system for scholars, libraries, and historical societies that would enable digital content management, the creation of simple web pages, and more complex web exhibits based on that digital content.

Omeka is standards-based, using the Dublin Core metadata standard to describe the items stored within it, and because Omeka is backed by a database, the data stored in it is indexed and searchable.

What is Neatline?
Neatline is a digital map-making framework that makes it possible to create rich, interactive maps and publish them to the web. Developed by the Scholars' Lab at the University of Virginia library, Neatline is built as a collection of mix-and-match plugins for Omeka. Neatline adds an interactive editing environment that allows users to bring together modern geography base layers, high-fidelity vector annotation, georectified historical maps, and timelines.

About the Workshop
Today we'll take a look at the basic features of Omeka, along with recommendations for best practices in creating and managing digital assets within it, and we'll look at the basic feature set available in Neatline.

We'll walk through the steps of building a simple interactive map in Neatline: Importing georeferenced maps, importing an Omeka item to a Neatline item, creating a collection of records and plotting them on the map, and adding interactions that make it possible to reason about change over time.
Part 1: Omeka

The Omeka Dashboard
When you first log in to Omeka, you’ll see the Omeka Dashboard, which gives you a quick snapshot of recent activity.

On the very top of the screen is a menu bar that tells you the name of the Omeka site you’re logged in to on the upper left corner. If you have administrator rights to the site, on the right side of the top bar is a menu of items that control the Omeka site itself. The options there include Plugins (to install/uninstall, activate/deactivate, or configure plugins that you’ve added to the Omeka site), Appearance (where you can select a new theme for the public display), Users (where you can add, modify, or delete user accounts from the site), and Settings (where you can adjust overall settings for the site such as its title, the email address for the site administrator, copyright and author information, etc.)

In the center of the Dashboard, you’ll see the snapshot of your Omeka site including, depending on your user type, total numbers of items, collections, tags, plugins, user accounts, the current theme, and the number of exhibits. Lists of Recent Items and Recent Collections are also displayed here.

On the left side of the center block, the main menu for working with your digital content is displayed. From here you can select Items (to view the list and edit groups of or individual items), Collections (to view and edit the collections on your site), Item Types (to view the list of item types defined on your site and the
number of items assigned to that type), Tags (to view and edit the list of tags used on your site), and Exhibits (to view and edit the exhibits defined for your Omeka site).

Depending on the plugins that are installed and the user capabilities that are assigned to you, you may see other menu items on the left side menu bar. Typically, here at UVa we install three additional plugins: Neatline (from which access the Neatline exhibit browse screen and all other Neatline functionality), Simple Pages (which gives you the capability of creating a simple HTML interface for your Omeka/Neatline site), and Bulk Users (which allows the addition of multiple user accounts in a single operation).

Omeka Items
The basic unit of an Omeka archive is the item. Each item record represents one digital asset in your Omeka archive.

Knowing what you wish to present about your items, the story that you want to tell using those items, is key to designing your archive. By consistently describing your items, you can build a cohesive archive, which illustrates your research and analysis. Consistent description also allows viewers of your digital archive to search for and retrieve items efficiently.

Dublin Core Metadata
"The Dublin Core Metadata Element Set is a vocabulary of fifteen properties for use in resource description. The name "Dublin" is due to its origin at a 1995 invitational workshop in Dublin, Ohio; "core" because its elements are broad and generic, usable for describing a wide range of resources." (from http://dublincore.org/documents/dces/)

The fifteen basic elements are fully defined, with examples of their use, at http://dublincore.org/documents/usageguide/elements.shtml

All information in the attached dictionary was taken directly from that page.

Tags and Collections
Tags and Collections are the means by which you organize the items in your archive for display, search, and retrieval.

Each item can have as many tags as you wish. Tags on an item can be added, edited, or deleted at any time.
Each item may only be assigned to one collection at a time. Collection membership can always be edited for any item.

**Creating an Item**
For this workshop, choose an image that you’d like to upload to Omeka.

1. Click the **Items** menu button on the left side menu bar.

2. Click the green **Add an Item** button. This opens the Add an Item data entry form, which has four sections:
   a. **Dublin Core** (where you enter metadata about your item)
   b. **Item Type Metadata** (where you may enter metadata specific to the item type you define for the item you’re adding)
   c. **Files** (where you upload your data files to Omeka)
   d. **Tags** (where you add tags that help to group items together by theme or topic)

3. On the **Dublin Core** section, fill in a **Title**, **Description**, **Creator**, and **Date** for your digital image.

4. Click the **Item Type Metadata** link to switch to that section.

5. Select the item type that best matches the subject of your digital image from the drop down list and fill in any additional metadata that you wish to record for your item.
6. Click the *Files* link to switch to that section.

7. Click the **Choose File** button, and navigate to the location of the digital file on your local or network disk drive, select the file, and click **Open**.

8. Click the *Tags* link to switch to that section.

9. Add tags that describe your item, separating individual tags with a comma. When you've finished listing your tags, click the **Add Tags** button to complete the addition.
10. When your data entry is complete, you can click the *Public* checkbox to make the item publicly viewable and/or the *Featured* checkbox to set the item as a featured item on your Omeka front page, if desired, or you can select a Collection from the drop down list to which to add your item. Then click the green **Add Item** button to save the item to your Omeka archive.
Part 2: Neatline

Introduction
Neatline is a plugin that adds geotemporal functionality to an Omeka archive. Using Neatline allows you to tell visual stories with maps and timelines that interpret single items or a collection of items or cultural objects.

Neatline Plugins
Neatline is a plugin for the Omeka exhibit builder and content management system, but there are plugins for Neatline that extend its functionality. Neatline Simile adds timeline capabilities to Neatline exhibits. Neatline Waypoints adds contextual reference points that can help guide viewers through complex sequences of events. And Neatline Text allows linking a long form narrative to annotations in Neatline exhibits.

The Neatline Browse Exhibits Screen
When you click on the Neatline menu button on the left side of the Omeka dashboard, the Neatline Browse Exhibits screen opens. This screen shows the complete list of Neatline exhibits in the current Omeka instance. From this screen, you can create a new exhibit, link to the public or fullscreen views of a Neatline exhibit, modify an exhibits setting, or delete an exhibit, if your user permissions allow.
Creating an Exhibit

1. Click on the **Neatline** tab at the bottom of the vertical menu on the left.

2. Click on the **Create an Exhibit** link at the top of the screen. This opens up a form where you can enter basic information about the new exhibit.

3. Enter a title into the **Title** field. Since we’re all creating separate exhibits, include your name in the title so that it’s easy to find your exhibit when it’s listed with everyone else’s.

4. For now, don’t worry about the URL Slug (the system will automatically generate a value based on your title) or the Narrative field. Scroll down to the **Default Layer** dropdown and choose the **Open Street Map** and **Google Hybrid** options.

5. Click **Save Exhibit**. You’ll be taken back to the main exhibits browsing interface, where you’ll see a new listing for your exhibit (depending on how many exhibits have been created, you might have to click through to the second or third page).

Once you find your exhibit, click on the exhibit title to launch the Neatline editor:
Before we start adding content, we need to set the default spatial location and zoom level that’s automatically displayed when the exhibit starts.

**Set the Default Map Focus:**

1. Start by panning and zooming the map to exactly the position that you want to use as the default. Since we’re going to be working with maps of Charlottesville, Virginia, click and drag on the map to focus on central Virginia, and then zoom in far enough that the city takes up most of the viewport.

2. Once you have the map in the right place, click on the **Styles** tab at the top of the editing panel on the left side of the screen, and scroll down to the **Default Map Focus** and **Default Map Zoom** fields. Click the **Use Current Viewport as Default** button, and the fields will be automatically populated by the current latitude/longitude focus and zoom level of the map.

3. Click on **Save** to lock the new defaults. Now, when you refresh the page (or when a public visitor comes to the exhibit), the map will automatically start at this location.

Now, let’s add some content. We’ll start by importing an historic map of Charlottesville.
Import the Historic Charlottesville Map

1. Click back to the **Records** tab at the top of the screen, and then click on the **New Record** link.

2. In the **Title** field, enter “Charlottesville - 1964.”

3. Click on the **Style** tab at the top of the screen and scroll down to the **Imagery** field set. In the **WMS Address** field, enter:

   [http://maps.neatline-uva.org/geoserver/wms](http://maps.neatline-uva.org/geoserver/wms)

   And in the **WMS Layers** field enter:

   **neatline:Charlottesville_rec_v2**

4. Click **Save** at the bottom of the form, and the map should appear as a translucent overlay on top of the satellite imagery.

5. Next, we’ll want to bump up the opacity of the layer so we can see it better. Scroll up to the **Opacities** field set and change **Fill Opacity** to ~95. You can type directly into the form input, and you can also click and drag up or down on the page to smoothly change the value of the input.

6. Finally, we’ll add one more piece of information to the record that will make the map play nicely when we start adding other items to the exhibit. Scroll down to the **Dimensions** field set, and enter “1” in the **Order / Weight** field.

   This essentially sets a vertical “stacking” order for the map that will allow us to control whether it displays above or below other content in the exhibit.

7. Click the **Save** button at the bottom of the exhibit. Go back to the list of records by clicking the **X** button at the top of the form.
Now that we have the map layered on top of the satellite imagery, let’s experiment with some of Neatline’s basic vector-drawing capabilities by adding some simple annotations to some of the recognizable waypoints on the map.

**Plot the Rotunda at the University of Virginia**

Let’s start out by dropping a point to mark the location of the Rotunda on the campus of the University of Virginia.

1. Pan the map west of Charlottesville until you find the Rotunda. Click on **New Record** and enter “The Rotunda” in the **Title** field.

2. Select the **Map** tab to display the geometry editing controls. Activate the **Draw Point** option in the list of editing modes.

3. Click once over the Rotunda to lay down a single point. Once the point is in place, reactivate the default **Navigate** mode so that you don’t accidentally drop another point the next time you click on the map.

4. Switch to the **Style** tab and scroll down to the **Dimensions** field set. Click on the **Point Radius** input and drag upwards to make the point a bit larger.

5. Once you’re done, click **Save** and go back to the list of records by clicking the **X** at the top of the form.

Next, let’s use the line-drawing tool to outline the course of the Rivanna River to the east of the town.

**Outline the Rivanna River**

1. Pan the map to the east of Charlottesville and find the Rivanna River, which runs north-south near the left edge of the map. Click **New Record** and enter “Rivanna River” in the **Title** field.

2. Click the **Map** tab and select **Draw Line**. Once the control is active, you can draw segmented lines on the map by clicking at
the locations where you want the points to be. Once all the points are in
place, you can “close” the line by double-clicking at the location of the final
point (and it’s always possible to modify geometries after the fact, so don’t
worry about being too precise on the first pass.)

In this case, start out by clicking once at the northern end of the river, and
then move the cursor down and double-click at the point where it leaves the
map to the south. This will draw a straight line between the two points.

3. Now, we’ll trace out the actual path of the river by
successively bisecting the line and dragging the points into
position on the river. Click on the Modify Shape radio button
and then click on the line that you just created on the map.
Three control points will appear – two at the ends of the line
(the points that we’ve already added) and a third, more
transparent point in the middle of the line.

4. Click on the middle point and split the line into two segments
by dragging the point towards the sharp bend near the middle
of the map.

5. Continue bisecting the segments and refining the path until
you have a good outline of the river. If you make a mistake and
want to get rid of a point, hover the cursor over the point and
press the “D” button on the keyboard.

6. When you’re finished, click the Save button at the bottom of the form.

7. Now we have the path in position, but it’s really hard
to see because the default line color (black) blends in
to the color of the river on the map. To fix that, click
over to the Style tab and scroll down to the Colors
field set. Click on the Stroke Color input to open a
color-picker widget and choose some sort of bluish
hue that goes well with the surrounding color
scheme on the map.

8. Let’s also make the line a bit thicker so that it’s more
noticeable. Scroll down to the Stroke Width input
and drag the value up to around 5 or 6.
9. As always, click **Save** to commit the new data, and head back to the list of records by clicking the X button at the top of the form.

Next, let’s use the polygon-drawing tool to annotate the graphical representation of Monticello on the map.

**Draw a Polygon Around Monticello**

1. Create a new record called “Monticello.” Pan to the far right corner of the map and zoom in to focus on the area near Jefferson’s home.

2. Click on the **Map** tab and activate the **Draw Polygon** mode. This works just like the line tool, except that the points are automatically linked together into a closed polygon as soon as you lay down more than two points. Start out by drawing a really rough shape around Monticello, even just a simple rectangle.

3. In the same way that we progressively refined the line to match the course of the Rivanna River, activate the **Modify Shape** mode, click on the polygon, and start refining and adding vertices to smooth out the shape.

4. ![Image of polygon drawing process](image)

   you’re happy with the polygon, head over to the **Style** tab and scroll down to the **Opacities** field set. Drop down the **Fill Opacity** to about 15%, and change the **Select Opacity** to around 20%. This leaves the outline of the shape unchanged but makes it easier to see through the drawing underneath.

5. Click **Save** to commit the changes.

**Add an Image and Text Description**

So far, we’ve just been adding vector geometries with simple titles to the map. Neatline also makes it possible to add images, long-format prose narrative, embedded audio and video files, and any other type of information that can be
represented as HTML. Let's add some descriptive information to our polygon outline around Monticello.

1. Click on the **Text** tab. Next to the label or the **Body** field, click the **Edit HTML** link to open up a fullscreen rich text editor.

2. To get some testing content, open up a new tab and go to the Wikipedia page for Monticello - [http://en.wikipedia.org/wiki/Monticello](http://en.wikipedia.org/wiki/Monticello). Click on the main picture of the building at the top left of the page, and then right click on the larger version of the photo. Click on “Copy Image URL” and switch back into the tab with the Neatline editor.

3. Click on the **Image** button and paste the image location into the **URL** input.

4. Delete the value that gets automatically populated into the **Width** input (this can be useful if you want really precise control over the layout of the image, but for now we'll just let the browser do what it thinks is best).

5. Click **OK** to insert the image into the document.

6. Head back over to the Wikipedia page for Monticello and copy the first three paragraphs of the article. Switch back into the editor and paste the content under the image, just like you would in a desktop text editor like Microsoft Word.

7. Click the **Minimize** button to close the editor. Notice that Neatline has copied the generated HTML markup from the editor into the **Body** field.

8. Click **Save** to lock in the changes and close the form.
Add a Timeline to the Exhibit

1. Click the **Back to Omeka** link at the top of the content management pane, above the title of the exhibit.

2. Find your exhibit in the list and click on the **Exhibit Settings** link to open the same form that we originally used to create the exhibit.

3. Scroll down to the **Widgets** field. Click on the input and select the **SIMILE Timeline** option.

4. Click **Save Exhibit**, which will take you back to the browse exhibits view. Click on the title of your exhibit to re-open the Neatline editor.

Next, let’s customize the timeline’s default focus date and zoom level.

Configure the Timeline

1. Click on the **Plugins** dropdown at the top of the editing panel and select the **SIMILE Timeline** option.

2. Enter “1957” into the **Default Date** field. This is the date that the timeline will automatically focus on when the exhibit starts.

3. Select the “Year” option in the **Interval Unit** dropdown. This effectively sets the zoom level of the timeline – the unit of time used to delimit the horizontal axis.

4. Drag the **Interval Pixels** input down to around 80. This determines the width between the tick marks representing time units selected in the last step.

5. Finally, set **Tape Height** to 25. This sets the vertical height of the “span” graphics used to represent duration events on the timeline.
Plot the 1964 Map on the Timeline

Now that the timeline is loaded and configured, let’s use it to control the visibility of the 1964 Charlottesville map.

1. Open the “Charlottesville - 1964” record, click on the Styles tab, and scroll down to the Dates field set.

2. Enter “1957” into Start Date and “1965” into End Date.

3. Now go back to the top of the form and find the Widgets multi-select input. Add the record to the timeline by clicking on the listing for SIMILE Timeline.

4. Click Save at the bottom of the form. The map should appear as a span between 1957 and 1965 on the timeline.

5. Next, let’s configure the record so that it’s only visible on the map between those two dates (right now, the record is plotted on the timeline between 1957 and 1965, but it will remain visible on the map regardless of where the timeline is focused). Go down to the Dates field set and enter “1957” in the After Date input and “1965” in the Before Date field. Now, the map will only be visible when the “center” line on the timeline falls between 1957 and 1965.

6. Click Save and close the form.

Lastly, let’s plot Lane High School and Venable Elementary School on the map, and configure them to appear on the timeline and react according to movement of the timeline.

Plot Lane High School and Venable Elementary School on the Map

1. Create a new record called “Civil Rights Era School Closings.” Zoom in to locate Lane High School and Venable Elementary School just east of the University.
2. Click on the **Map** tab and activate the **Draw Polygon** mode. Draw polygons around both schools.

3. Activate the **Modify Shape** mode, if desired, and click on the polygons to start refining and adding vertices to create the shape you desire.

4. Once you’re happy with the polygons, head over to the **Style** tab and set the fill color and opacity values as you desire.

5. Still on the **Style** tab, scroll down to the dates fields. Governor Almond closed these two schools in September of 1958 to resist enforcing desegregation. Let's plot both **Start** and **End** dates, and **After** and **Before** dates for this record. This will create a tape on the timeline, and will allow the map annotations to appear and disappear as the timeline shifts.

   In the **Start Date** field, enter 1958-09-19 to encode the closing date of September 19, 1958. In the **End Date** field, enter 1959-01 to encode the reopening date of January 1959.

   In the **After Date** field enter 1958-09-19, and in the **Before Date** field enter 1959-01. This will ensure the annotations are only visible between these dates.

6. On the **Style** tab, click in the Widgets field to see the list of available widgets, and select SIMILE Timeline.

7. 

8. Click **Save** to commit the changes.

Scroll the timeline back and forth to look at how the annotations and the map appear and disappear in relation to the date that’s centered on the timeline.

**Adding an Omeka Item to the Neatline Exhibit**

Now that we’ve spent some time with adding vector annotations to the Charlottesville – 1964 map, let’s add our Omeka item as a Neatline item.
Add the Omeka Item as a Neatline Item

1. Add a new record called “Photo”

2. Go to the Item tab and click the Search Omeka items drop down menu. Select your photo from the list, or search to find it by title.

3. Once your item is selected, the Dublin Core Metadata information you entered about the image will be displayed.

4. Click on the Map tab to create an annotation on the map.

5. Click on the drawing tool that best suits the annotation you wish to make.

6. Style the shape on your map using the tools on the Style tab.

7. Click the Text tab, and in the Body input box, add any text you wish to more fully describe the image you’ve added to the Neatline item. This text will appear at the top of the pop-up for the annotation at the top of the imported Dublin Core metadata from the Omeka record. Style it as desired, using the HTML editor as in the previous exercise.

8. Click Save.
Civil Rights Era School Closings
Governor Almond closes Lana High School and Venable Elementary School rather than allow them to be...
Monticello
Riveranna River
Rotunda
Charlottesville - 1964
Prerequisites

- At least 8GB of free space on your computer’s hard drive.
- Multi-core processor (pretty much any reasonably current computer should be ok).
- 2GB RAM minimum (4GB or more is better). This memory will be shared between the virtual machine, your computer’s main operating system (Windows/Mac OS), and any applications you are running under Windows/Mac OS.
- The more resources you can dedicate to the virtual machine, the better it will run. As soon as you shut down the virtual machine, those resources will be freed up and become available to your main operating system.

If you are uncertain about your computer’s specifications, feel free to contact me.

Install Software

- Download the appropriate version of VirtualBox for your operating system from https://www.virtualbox.org/wiki/Downloads.
  - Windows: http://download.virtualbox.org/virtualbox/5.1.22/VirtualBox-5.1.22-115126-Win.exe
  - Mac OS: http://download.virtualbox.org/virtualbox/5.1.22/VirtualBox-5.1.22-115126-OSX.dmg
  - Linux: https://www.virtualbox.org/wiki/Linux_Downloads
- Install VirtualBox.

Download Disk Image

- Download the virtual disk image (.vdi) file to your computer and save it to a location where it won’t be accidentally deleted.

Setting Up the Virtual Machine

- Start VirtualBox. You should see a screen similar to the one below, except that the list of virtual machines on the left side should be empty in your case.
- Click on the “New” button in the upper right corner. You will then get a pop-up window that lets you name the new virtual machine and select an operating system:
Select a name that’s meaningful to you. Under “Type,” select “Linux.” If VirtualBox doesn’t automatically choose it for you, select “Ubuntu (64-bit)” from the “Version” menu. (The virtual machine uses a Linux version called Lubuntu, but it is a derivative of Ubuntu, so that’s ok.) Then click “Next.”

- On the next screen, you can select the amount of RAM (working memory) that the virtual machine can take up. You should assign at least 1024MB (1GB). If your computer has 4GB, 1536MB or even 2048MB would be a good value. If you have 8GB or more, I would use 50% of the total memory available. Then click “Next.”
The next step is to select a virtual hard disk for your virtual machine. Select the third option ("Use an existing virtual hard disk file") and then select the .vdi file you downloaded. Then click "Create":
You should now have a fresh virtual machine show up in the left panel of the VirtualBox main window. There is one more thing you can do to improve your virtual machine’s performance (if your computer has the necessary resources). Select the virtual machine from the list and click on the “Settings” button at the top. You should see the following window:
• From the menu on the left, select “System”: 
On the "Systems" page, select the "Processor" tab:
Here, you can select how many processor cores will be reserved for the virtual machine. (Although your computer usually only has 1 processor, it would have 2 or more processor cores.) VirtualBox will indicate how many processor cores you can safely reserve for the virtual machine without negatively impacting the performance of your computer too much. VirtualBox indicates this by dividing the “Processor(s)” slider into a green and a red zone. In the example above, VirtualBox recommends that I can assign up to 4 (out of a total of 8) processor cores to the virtual machine. (I could assign more, but my main operating system [Windows 10] might be slowed down.) Select an appropriate value and click on “OK.”

- Now that you have a virtual machine, let’s use it. Select the virtual machine and click on the “Start” button at the top. A new window will open, some text will scroll by, and you will finally get the login screen for the virtual machine:
• Enter the password ("omeka") and click on "Log In." You will then see the Lubuntu desktop:
• Unless you want to edit Omeka’s application files, you will be doing all your work in the web browser (in this case, Mozilla Firefox). There are three bookmarks in the browser’s bookmark toolbar: the local Omeka admin page, the public homepage of your Omeka install, and the official website:
• You can log into the admin interface using the username and password in the “pwd.txt” text file on the virtual machine’s desktop.
• You can access any applications installed on the virtual machine through the button in the bottom left corner of the screen (similar to the Windows “Start” button):
To shut down the virtual machine, click on the power button in the bottom right corner of the screen (next to the clock) and then select “Shutdown” from the window that pops up:
The virtual machine window will close after a couple of seconds and you will be back on the VirtualBox main window.