IA and RIAs – You Know More Than You Think You Do
by Adam Polansky

I’ve been working as an information architect for nearly 10 years, but it wasn’t until recently that I had the opportunity to work on the development of a rich Internet application or RIA. While I’d made some effort to get an understanding of what this might involve, like many things you can’t really get a clear idea what it’s like to do something until you actually do it.

I work for an online travel company which has a group led by our senior vice president of design. This group began as a kind of skunk-work tasked with looking for a new way to offer travel options within the context of experience as opposed to just pushing lists of inventory against a set of hard parameters like destination, dates and product type.

The result of their efforts was a proposal for a planning application aimed at people who might or might not be traveling in the near future. The emphasis would be on rich content such as photos and videos, community-provided content like reviews and journals, travel guide information, articles, tips and advice. Most importantly, the application would contain a wish list feature that would let visitors create and save multiple lists that they could print, send or share by publishing to a community environment. The idea targeted people who are dreaming about and planning versus booking their travel.

The proposal took the form of a Macromedia Flash demo which wove a story that absolutely captivated everyone who saw it - our parent company, our CEO, CMO, marketing teams, supplier partners and the press. As a result the team was able to secure funding and resources to build a beta site. This was a big deal since we aren’t known for beta sites as our market is highly competitive and very quick to respond when competitors introduce new features. While all this was going on, I was involved in different projects and some internal strategic initiatives when my manager stopped by to tell me that the IA assigned to the planning application was going to follow another opportunity within the company and that there were some difficulties with the project and that I was needed to take over as IA.

The Landscape

It had been a while since I was dedicated to a single initiative and, as I mentioned, I’d somehow managed to go for years without ever working on an RIA. Here are some of the thoughts I had going in to the project:

High profile. The product owner had done an excellent job stirring up interest and convincing several people that we had an opportunity to differentiate ourselves in a market that sells largely commoditized products.

High design value. The fact that everyone had seen this interactive demo created a perception that the final product was much closer to being a reality than it really was.

Work from predecessor. I didn’t know how much or how little my predecessor had done. I was joining the conversation late and didn’t know how much I’d need to consume from him. Also, he was a particular expert in social web applications, which I knew was going to be a core attribute of the application. Up to this point, that had not been anything on which I’d focused; in fact, my own tendencies could be described as “anti-social web.” I’m of the opinion that nobody really wants to see or read about all my little mundane doings, and, for that matter, I wasn’t keen to share them.

RIA vs. HTML. I had no experience with RIA development and didn’t know how much of a knowledge gap I’d have to cross.

The hammer problem. There’s a saying, “When the only tool you have is a hammer, every problem looks like a nail.” Without knowing my place on the learning curve, I didn’t want to be guilty of trying to shoehorn processes into my personal “safe-zone.”

Adam Polansky is principal information architect with an online travel company where he concentrates on Internet applications and processes associated with the practice of IA. He is a contributing author to Usability Success Stories – How Organizations Improve by Making Easier-To-Use Software and Web Sites. The author can be reached by email at apolansky<at>redearthia.com.
So, being a good IA, the first thing I did when I joined the team was shut up, listen, try to get a handle on the state of the project, see what obstacles or pain-points were creating problems and then figure out where my services were needed. At a strategic level, here’s what I observed:

Limited prototype demo. The demo was built around the particular needs and attributes of one destination – Las Vegas. Several issues around scalability had been identified but not addressed, including the ability to roll cities up into regions like the Caribbean and paths that would be predicated on an activity like skiing versus a specific geographic location.

Existing work by predecessor. Very little, if any, work had been done. My predecessor had been split between two engagements, one of which grew into the new opportunity that he chose to follow.

Maintenance. The methods of content and digital asset management were not scalable to the demands that this application would need over time, either in terms of rolling out new destinations or maintaining existing ones.

Urgency. Everyone wants it now, from our parent company to the press. This contributed friction because there were still so many unknowns in getting this effort into production. Design led the way up to this point.

From a tactical standpoint there were additional considerations:

Scope. A detailed list of features and functions had not been put through a qualification process to determine priorities on anything other than stakeholder preference.

Usability. The design that had been so aggressively marketed had not undergone any scrutiny from a usability standpoint.

Content. Several content areas had not been fleshed out and the need for interaction design prior to committing to code still existed.

Navigation. The proposed list of initial destinations included different types of paths that had not been distinguished from each other.

Back to the Foundations

Where to begin? Well, when there are this many issues at play all across the application, you begin by looking at the foundations.

Usability Testing. Fortunately, the team had already set time to get into the lab with the prototype, but the testing guidelines had yet to be outlined and refined. So the first thing to do was to get the prototype ready for the lab and identify what we wanted to observe. I wanted to see how a richer interface did or did not influence a participant’s ability to satisfactorily complete tasks. Did fancy graphics pardon functional sins? What attributes resonated most with users? Did the application do a better job of making trips seem real or feasible? Here’s what I was able to take away from those sessions.

Influence of images (again). We already knew from Jakob Nielsen’s Tips for Shopability [1] to “[e]nsure that images are big and show features that are important to buyers.” When participants saw the big photos and videos, the responses where immediate and favorable. Ideally, in a retail environment, you want people to experience your product. That’s why you put the remote in the hands of someone looking for a TV, you set garments out where people can handle them and you ask potential buyers to test-drive a car. This creates transference - the ability to see the product as your own. Well, in travel, you can’t send someone to the Caribbean for a couple of minutes to see how they like it, but you can use imagery to get people to imagine themselves poolside, in the restaurant or enjoying a show. We saw participants become extremely vocal about seeing themselves in the pictures.

Emphasis on qualitative vs. quantitative information. In spite of all the visual material, invariably everyone came down to the particulars. How much per night? Does the hotel allow animals? Does the hotel have Internet access? What time are the shows? No surprise here. It doesn’t take long for people to ask “Where’s the meat?” While they love being caught up in the interaction and the transitions between key-frames, sooner or later the rubber is going to meet the road and people remember why they’re on the site in the first place.

Pragmatic ammunition. The observations gave me the ammunition to go back into design and business meetings with proposals for refining the application, fortified with rationale for changes that might not be as exhilarating to the imagination but were based on common sense. This served to counter some of the giddiness surrounding the effort and to establish my voice as a group lead.

Feature Analysis. The application was a collection of ideas that existed mostly within the gestalt of the team.
couple of things. We needed to reach an agreement on what we felt were the logical resting points or key-frames between transitions. We also needed to create a way to communicate basic transitions within a static page.

The key-frames were fairly easy to distinguish because we were building in flags that our analytics system would use to return some basic click-through analysis. These key-frames were simply listed in a spreadsheet and served as an inventory of which content areas had been defined, planned and built.

However you choose to do it, qualifying feature and functions earlier rather than later in the project should be aimed at the following outcomes:

**Objectivity** – By leveraging individual bias to generate unbiased ranking of features, (by limiting participants to rating features only within the context of their areas of expertise and using over-riding constraints and the combining of scores to get a single rating), features couldn’t be pushed (at least, not easily) by personal agenda.

**Project planning** – Scope and estimates provide the basis for a traditional project Gantt chart or the backlog that will feed an Agile Iteration plan.

**Mitigating churn** – Greatly reduces the down-stream second-guessing in development when features have not been pre-qualified. There are fewer surprises.

New Content Areas. The prototype implied additional content areas for research information like “Insider Tips & Advice,” maps and community-driven content. There were images and links to show where you’d find this content but the interaction design associated with them still hadn’t been completed.

Here’s where it hit home that the need for good old low-risk modeling was alive and well in the rich application development world. In other words, we needed wireframes or something like them. It’s important to mention that this wasn’t my initial assessment but rather a request from the development team and the business ownership.

The first thing I did was have a discussion with the design team to agree on what we believed were the logical views to display in the wireframe. HTML lends itself easily to a set of pages, while Flash is essentially one page with a lot of internal transitions. This fluidity meant a
This called for some re-thinking of the basic navigation. A visitor to the site might begin by saying “I want to go to a particular place,” or they might begin by saying “I want to ski, and I don’t know where yet.” I developed a model to show how a visitor might be able to find information based on either premise, as shown in Figure 3.

Figure 3. Navigation options

By incorporating usability testing, feature analysis, semi-traditional interaction design and a dash of findability principles, I was able to contribute to the beta launch of what we expect to be a new direction for how we relate to our customers in ways that are relevant to them.

As of this writing the project is still in beta and coming into focus as we collect external comments and experiment with our own ideas.

What Did You Learn, Dorothy?

When it comes to the information architect’s role in RIA development, a few things are different. Most things are the same. Some things are the same - only more so.

What’s Different?

- **Vocabulary** – There is some language that comes with RIA development that is an outgrowth of its visual fluidity. This affects traditional artifacts.
- **Business logic in the user interface** – There is more going on in the UI user interface than there used to be. This informs the efficacy of certain interactions.
- **Interaction design context** – Interaction design often is driven at a component rather than a page level.
- **Revision cycles** – Lead-times between functional and visual design can be longer.
- **Hype** – There’s a higher giddiness factor that comes from stakeholders seeing cool shiny moving prototype.

What’s the Same?

- **The process** – Idea, plan and build
- **Planning** – There is still the need to do low-risk, functional prototypes devoid of design elements.
- **Tactics** – Assess your surroundings and choose tactics and deliverables appropriately.
- **Balance** – IA always needs to bring balance between the system and the UI. However, the desire to forge ahead without planning which used to come from technology is now coming from design.
- **Vision** – It takes discipline to keep the big picture in sight when developing individual features.
- **Collaboration** – The discipline of validating the feasibility of features with technical development is necessary so that a brilliant visual design doesn’t get built without any muscle behind it.

Pitfalls to Avoid

- **Revision cycles** – Don’t assume that visual edits are as easy in Flash as in HTML.
- **Tone-setting** – The IA’s job is to be a pragmatist. Projects like this live on enthusiasm. Be diplomatic with the team so as not to kill the excitement.
- **Interaction design** – Don’t get caught up in the interaction at the expense of content or the Rube Goldberg [5] school of interaction for its own sake.

Here’s the moral of the story: If IA efforts are the logical outgrowth of common sense, then the development platform won’t change too much about how IA is applied. Furthermore, if the IA’s job is also to assess an environment and decide on an appropriate approach to the problems, then nothing is different.

**Resources**

[3] Gantt chart: popular type of bar chart that illustrates a project schedule. For additional information see http://en.wikipedia.org/wiki/Gantt_chart
[5] Rube Goldberg was a comic artist who created all the screwball inventions in the old Warner Brothers cartoons that used as many steps as possible to do something very simple. For more information see, for instance, www.rgmc.com/ or http://en.wikipedia.org/wiki/Rube_Goldberg.