Adobe had its own Apple moment today, announcing a suite of new products in the same venue, San Francisco's Yerba Buena Center, that Tim Cook used to premiere the iPhone 5. Adobe's announcements won't capture the public's imagination like a new iPhone, but these tools will have a profound impact on what the public sees on their iPhones—and all the other screens in their lives.

At today's "Create The Web" event, Adobe introduced an integrated set of tools designed to tame the complexity of building the modern, multi-screen web—without obscuring the underlying code. In other words, this is not your dad's Dreamweaver, not just new faces on established Adobe software. These
are ground-up, web-centric tools for visually manipulating the parameters of the standard HTML5, CSS and JavaScript code that the contemporary web is made out of.

This has been coming for a couple of years, but the emergence of the Adobe Edge Tools, as these new products are called, represents an incredible departure from the company’s previous products and the way they are developed. In short, Adobe has embraced the open web, contributing input and code to both WC3 web standards and WebKit open source web browser engine. Even more importantly, it has been developing these new products the way all the open source kids do on GitHub, releasing builds early and often and incorporating feedback and in some cases open source contributions from the developer community.

It is a tricky rode to walk, as the recent flap about the open source status of MakerBot’s latest 3D printer demonstrated, but Adobe looks to be treading both carefully and confidently. If they succeed, they will have done a near impossible thing: to merge open source development practices and for-profit products without users feeling ripped-off.

This is far from a fait accompli, but if it works it will be because what Adobe is offering to designers and developers is so generous, so reasonably priced and so respectful of the mores of the various tech tribes that it’s hard to feel ripped off. Of course, in a world where everything on the web can be had (legally or not) for free, actually paying a monthly fee of $49.99 (or less) to Adobe, could rub some users the wrong way. But if you can actually use the tools to do paying work, it’s a very modest cost that is easily justified.

The astounding fact is that for that monthly fee (lowered to $29.99 for students, teachers and previous Adobe Creative Suite users) you get access to the latest versions of all of Adobe’s software products, the moment they are released. In the fast-paced world of web development, this is a very good thing.

I am a long-time Creative Suite user, and I have been testing out Creative Cloud over the past few months. The new versions of the standard desktop tools, Photoshop, InDesign and Illustrator are all faster, smoother and more powerful. Dreamweaver, Adobe’s all-in-one code editor/website builder, on the other hand is still hard for me to get comfortable with. I really enjoy using Coda as a text editor, and I know many developers who swear by TextMate.

And although you can build responsive sites in the new Dreamweaver, I found the proscribed views assigned to desktop, tablet and mobile to be arbitrary and inflexible. So as I was watching the stream of the product announcement today and looking at the descriptions of the Adobe Edge products on the web, I was thinking, “That’s more like it!”

A big part of the quickening of this new, open source and subscription orientation, came a year ago when Adobe acquired Typekit, the company that finally brought the full spectrum of typography to the web. Several companies have emerged to serve web fonts, but Typekit did it first and has built the
easiest to use and most reasonably priced platform. When Adobe bought Typekit, they also got its CEO and founder, Jeffrey Veen. Veen is still running Typekit, but also oversees Creative Cloud, and his touch is clear. People familiar with both companies have told me that Adobe bought Typekit to become more like it, and that, indeed, seems to be the case.

The Adobe & HTML website is really a clearing house for all of the issues that preoccupy web designers and developers, and the way that Adobe has inserted itself into helping to solve and provide tools for these issues creates confidence in the whole operation. What’s more, many of these tools are built with the same web standard code that they are capable of manipulating. Here is a quick rundown of the new tools with some note of their relevance and state of release:

Adobe Edge Animate effectively replaces Flash as a timeline based rich media composer, but all rendered in HTML5, CSS and JavaScript. All the wacky, animated stuff you used to need Flash for you can now do in Edge Animate. Advertisers and marketers will be especially interested since these animations are scalable, enabling banner ads, for example, to resize dynamically to different screen sizes.
Adobe Edge Reflow builds upon the embryonic responsive design tools in the current Dreamweaver and gives you complete control over breakpoints between different screen sizes, as well as WYSIWYG control over CSS styles and the ability to identify and export all of the relevant bits of code. Reflow is not quite ready for release, but will be a big boon to multi-screen designers (which soon will mean all designers) when it is.

Adobe Edge Code is the first, true contextual text editor. Built on top of the Adobe-sponsored open source project Brackets, Edge Code will seriously make me consider putting aside Coda. Web development is an inherently messy business in which a single web page is composed through reference to many individual CSS and Javascript files. What Edge Code does is enable you to edit the relevant bit of the relevant file right from the place in the html code that references them. Not only is there code hinting, but inline color pickers and font selectors, as well. And you can see the results of your changes with a live WebKit preview. Believe me, this will save developers a lot of time and make designers much more comfortable touching code. An experimental build is available for download now and more features will likely be added before an official release.
Adobe Edge Inspect is the full release of the product (which I wrote about this summer) that started out as Adobe Shadow. Edge Inspect solves the incredibly time-consuming problem of previewing a design on multiple devices. With Edge Inspect, you load an app onto each device you want to test which allows your desktop machine to push the content on its screen to those devices. Not only that, but you can inspect the code on any mobile device as if it were your desktop browser, which takes a lot of the guesswork out of debugging mobile browsers. And a really helpful little bonus, you can instruct the devices to all take screenshots simultaneously and place them on your desktop to make emailed client presentations a snap. This is ready to go now.

Adobe Edge Web Fonts and Typekit make using all manner of fonts on the web easy and in some cases free. Along with building their own service (which now includes access to 1,500 fonts with another 1,000 classic fonts soon to come through an agreement with Monotype) Typekit has also been collaborating with Google for the last couple of years on its web font APIs. The result is a new product, Adobe Edge Web Fonts, that will take all of Google’s fonts and add a selection of Typekit’s fonts (for a total, to start, of 500 fonts) and make them completely free to use on the web—with no restrictions. This
is an example of the open source generosity I mentioned above. Jeffrey Veen
told the crowd today that he believes not only in making more fonts available
through Typekit, but more fonts on the web altogether. 500 free, high-quality
fonts with simplified code requirements is certainly a great place to start.

Adobe PhoneGap Build is another time-saver and expertise reducer. Adobe
also bought PhoneGap recently, a company that makes software that “wraps”
standard web code to make it into mobile apps that behave in most ways like
native apps. The additional innovation here is that the “Build” aspect means
that PhoneGap can package an app for different platforms (iOS, Android, etc.)
without the developer having to set up a dedicated environment on their
computer for each of those platforms. Even more than PhoneGap itself, the
ability to compile apps in the cloud will lower the barrier for entry into the
mobile market considerably. PhoneGap Build is available now as part of the
Creative Cloud.

Will these tools change the way that we “Create The Web”? Very likely. The
ability to manage all of the bits and pieces of web projects and be able to
visually tweak every bit of code will, over time, turn non-coders into coders.
And coders may not become better designers (many developers just
wish somebody else would do the design) but they will certainly become more
productive and have more confidence in doing things they have never done
before. The result of all of it will be better looking content on all our devices,
which should prompt even better tools from Adobe and others for making
even more.

To keep up with Quantum of Content, please subscribe to my updates
on Facebook or follow me on Twitter.

This article is available online at:
But you can’t play Flash content on an iPhone! Well, we can cross that argument off the mobile decision grid. According to a scoop last night on ZDNet, Adobe has announced that they are stopping development on Flash Player for mobile browsers. “Our future work with Flash on mobile devices will be focused on enabling Flash developers to package native apps with Adobe AIR for all the major app stores. We will no longer adapt Flash Player for mobile devices to new browser, OS version or device configurations.”

Adobe is, in fact, finally heeding Steve Jobs’ advice. In a public letter about Flash last year, Jobs said, “Even if iPhones, iPods and iPads ran Flash, it would not solve the problem that most Flash websites need to be rewritten to support touch-based devices. ... New open standards created in the mobile era, such as HTML5, will win on mobile devices (and PCs too). Perhaps Adobe should focus more on creating great HTML5 tools for the future, and less on criticizing Apple for leaving the past behind.”

Adobe is, of course, aiming for a positive spin, on an official Adobe blog, Danny Winokur, VP & GM for Interactive Development, wrote, “Over the past two years, we’ve delivered Flash Player for mobile browsers and brought the
full expressiveness of the web to many mobile devices. ... However, HTML5 is now universally supported on major mobile devices, in some cases exclusively. This makes HTML5 the best solution for creating and deploying content in the browser across mobile platforms.”

On Sitepoint’s designfestival, Joel Falconer points out that this is all part of a larger strategy by Adobe to be a player in the world of HTML5 and cites their recent acquisition of the web font service Typekit and the release of a flash to HTML5 converter as evidence of the shift.

For those of us who have felt that HTML5 was clearly the future, we no longer have to ask ourselves, “but what about Flash?” It’s now clear that Flash has a role behind the scenes in game development and high end app creation—which seems like a good place for a proprietary technology to be. As HTML5 and the open web become increasingly app-like, proprietary (and paid) native apps will continue to be in the vanguard of mobile development because of their ability to integrate with ever-evolving hardware capabilities. True to form, Adobe is playing all the horses.

This article is available online at:
These days, everybody's got an idea for an app, and every app is a potential startup business. And yet, for the 755 apps that get submitted to the Apple app store a day, there are probably thousands that never get built.

And even for apps that get built, most never become actual businesses. Why not?

That is the question that General Assembly was designed to address. Based in New York City, GA defines itself as “a campus for technology, design, and
entrepreneurship, [that provides] educational programming, space, and support to facilitate collaborative practices and learning opportunities across a community inspired by the entrepreneurial experience.” GA has active campuses, as well, in London, Sydney, San Francisco and Melbourne with upcoming locations in Toronto, Berlin, Philadelphia and Boston and more to be added in the future.

What GA has done is to create a truly flexible, responsive—and affordable—professional education platform that supports all aspects of tech startups. The free-form curriculum itself clues you into all of the things you need to consider to make a startup work (there are no degrees, or at least, not yet). But beyond learning what you need to know, students also learn about the startup community of their own city (each location has a course dedicated to the subject) and meet others with similar or complementary skills. Similar to a co-working space, the General Assembly campuses are like co-learning spaces.

I had heard that Alexis Ohanian was an early investor, so I interrupted his work on his upcoming book, Without Your Permission, to ask him about the uniqueness of GA:

“"It was clear from everything going on that they were on to something and the timing couldn’t have been better — more and more people are seeing the opportunity a tech education provides for employment (or entrepreneurship!) and there aren’t many campuses that deliver both the actionable education and vibrant community that General Assembly does.

I went to a pretty great school, UVA, but even there you have a normal distribution of people where a standard deviation or two from the mean you get the overachievers (guilty as charged) spending a ton of time researching, studying, really trying to get the most out of their four years. A bunch are just having a good time and making sure to pass in four years.

On a campus like GA, there is a much higher concentration of people who are there not just to learn, but to make something people want (love) with others who do, too. That density of ambition and camaraderie isn’t just good for the education, but it builds a network and a community that continues to thrive long after “graduation.”

The learning opportunities are structured by commitment level, 60-90 minute classes, 2 hour–2 day hands on workshops, 8–16 week, twice weekly courses plus various events and online classes.

The goal is to jumpstart the startup communities in these places while building a library of online courses that can eventually serve an even wider audience.

I recently participated in Peter Bell’s excellent livestreamed online class, How to Build a Mobile App. I was curious about how GA would package this central topic to their audience into an online hour, and I was very impressed with the balance of survey and depth that Bell created.

In my headline I say, “an hour (or less).” So, if you’ve got an hour and you want to be conversant for your next cocktail party or elevator pitch, simply watch the video (recommended).

But, if you are impatient (and who isn’t on the web?) here is my quick
synopsis of the key points to help you get to the follow-up meeting with the VC.

**Why Mobile Matters:**

Some quick popcorn facts to make it clear that you understand the importance of mobile:

- **APPLE MAKES MORE IPHONES THAN HUMANS MAKE BABIES:** Apple moved over 37 million iPhones in the last three months of 2011, a higher rate of production than humanity’s rate of reproduction.
- **FACEBOOK ADDS 110 MILLION MOBILE USERS IN SIX MONTHS:** Now more than half of Facebook users accessing through mobile
- **AMAZON TOPS $1 BILLION IN MOBILE SALES IN PAST YEAR**

**Why Mobile Is Different:**

Bell makes it clear that unlike the desktop where we have a lot of assumptions about what a user is doing and where a user is interacting with our application, on mobile there are all kinds of “ad hoc” use cases beyond “on-the-go” including micro-tasking, local search and plain old boredom. Also, increasingly, mobile is the “second screen” that accompanies other “primary screen” experiences. In terms of how users interact with mobile applications, factors include not only screen size, but limitations of interaction (one eye, one thumb) not to mention attention span. So, in mobile, you have to be both ruthless about what constitutes a use case while supporting as wide a range of functionalities as possible.

**The Different Types Of Mobile Development:**

The main distinction in mobile strategy is between dedicated mobile sites and responsive/mobile first sites. In general, sites that have a very specific utility function are best conceived of as dedicated mobile sites. Bell uses the example of the United Airlines mobile app through which you can book flights and check on flight status, etc., and suggests JQuery Mobile is a useful javascript framework for building this kind of app.

Sites that are more document oriented and have a lot of pages to read can take advantage of responsive web design to create a “view” of a fully-featured website that is optimized for a variety of mobile screens. Bell uses the example of General Assembly’s own website as an example of this kind of site.

Once you decide whether you are making a stand-alone application or a view of your existing website, you have to decide how to turn a “site” into an “app.” Bell suggests three main methods, depending on the complexity and performance demands of what your app needs to do.

- **HTML 5 Based:** Simple apps can be built entirely in web standard HTML, CSS and JavaScript and then “wrapped” in code written by a tool called PhoneGap (a free service, bought last year by Adobe) to be delivered to any of the seven major mobile platforms. That same web code could power a “web app” that lives only on the web, but converting it with PhoneGap enables you to access some of a mobile device’s native features (i.e. location sensor or camera) and to upload it to app
stores which can boost your app’s discoverability.

- **Cross Platform Native:** If you need more of the look and feel of a native app without the cost of developing for multiple platforms, you can take that same web code and use a product called Appcelerator Titanium, which rewrites your app into native code. In the company's words, it “leverages over 5,000 APIs to create native iOS and Android apps, and HTML5 mobile web apps.” Wow!

- **Full Native Development:** Converting web code into applications is a shortcut that does have its limitations and tradeoffs. Bell recommends developing in native code (which is much more expensive and the developers of which are much more in demand) for apps that require high performance OR high data traffic. Only native development, at this point, gives you full control over server calls and memory management to avoid frustrating bottlenecks in user experience.

The great benefit of the hybrid approaches (using PhoneGap or Appcelerator vs native development) is that you can change things quicker and cheaper as you find what Marc Andreessen calls “product market fit.” Another benefit of HTML 5 based solutions is that they can be easily tailored for a wide range of screens as opposed to just iPhone, iPad or Android.

**Finding Your Startup Strategy:**

If you are trying to start developing your app on your own, you will need to at least get a handle on the basics of HTML/CSS and JavaScript and how to use PhoneGap or Appcelerator. If you can tell from what your app needs to do that you will need to go native, bite the bullet and learn one of the native tongues (Objective C for iOS or Java for Android.) Bell strongly advises that if you go the mobile route, pick iOS or Android and build for that first. Use the first platform to work out all of your preliminary kinks before building for the second platform. This is why you often see apps starting in iOS with Android “coming soon” (or vice versa.)

Understanding the differences between a web site with application functionality (a web app) and a dedicated app store ready app is important. This distinction will lead you to decide if you need a stand-alone website and a stand-alone app (with all of the extra development time and expense that entails) or if you can start with a responsive website that can do it all (with potential performance and device functionality penalties). From there you move on to the decision of whether to use PhoneGap, Appcelerator or Native code (choose one to start with!)

What mostly falls outside of the scope of Bell’s presentation are the complexities of backend (server-side) development. A whole hour could easily be spent on this topic, as well. Suffice to say there are languages like Ruby and Python that use frameworks like Rails (for Ruby) and Django (for Python) as well as end-to-end hosting solutions (like Brightcove’s AppCloud) that can be used to handle these things along with new solutions like Node.js that use JavaScript on the backend. The more complicated the data feeds are for your app, the more backend work required. Many apps use freely available API’s for data and can sidestep these complexities entirely. Understanding what data you need and what’s involved in getting it into your app should be part of your initial strategy decisions.
General Assembly: Learn To Talk Like A Mobile App Genius In Just One Hour (Or Less) – Forbes

So, that was about 20-minutes, right? Do you feel like a mobile app genius yet? If not, it’s all my fault. Please go back and watch Peter Bell’s telling of the tale.

And, if you happen to live in any of the cities that General Assembly has campuses in, show up and take classes. The single classes are generally $30, workshops $50-200 and full courses $3,000-5,000. The flexibility, quality and relevance of instruction can’t be beat. And, even better, they have a roster of eight upcoming online classes (including the mobile app class described here) coming up in the next month FOR FREE!

Also, I would encourage any city with an economic development program that wants to encourage tech startups (hello, Portland, Maine!) to set up a General Assembly campus. Beyond giving focus and training to your tech community, it will put you on the map for having one.

---

To keep up with Quantum of Content, please subscribe to my updates on Facebook or follow me on Twitter.

This article is available online at:
“Who knows what evil lurks in the hearts of mobiles? The Shadow knows!”
No, I’m not talking about the 30s radio serial, but an incredibly useful little product from Adobe debuting at SXSW Interactive today. The evil, in this case, is the cacophony of devices that modern websites need to serve.

With the rise of mobile traffic has come an increasing awareness on the part of site owners of the importance of addressing all those devices in a comprehensive way. Responsive web design is a flexible response to the problem. The key is to not expect your site to look exactly the same on all screen sizes, but to be willing to slide components around and resize elements in a fluid way so that your content look good in all possible screen dimensions.
Sounds great, right? The only problem is that not only are screen sizes different, but different devices have different browsers that render some page elements differently, as well. So, although the dream of HTML5 web standards and responsive web design is to create unified sites with a single code base, out in the wild it’s still a little wild—and getting more so.

So, what’s a responsible web developer to do? Most interactive shops that do mobile have a fleet of devices that they strew across conference tables and team members take turns testing them all. Pretty hit and miss. What’s more, when you actually find a problem, there’s a lot of guess work involved in figuring out what exactly in the code is the culprit. Most contemporary web browsers have a developer mode that lets you “inspect element” on any part of a web page on your desktop, but there hasn’t been any easy equivalent for mobile devices—until now.

Bruce Bowman, Senior Product Manager for Adobe’s Digital Media Business division, gave me an advance demo of Shadow and told me that extensive interviews with mobile web developers identified device testing as a huge “pain point” in their workflow. Many of the less experienced developers wanted a “device emulator in the sky,” that would enable them to test websites on a full range of virtual devices without having to physically possess them. But more experienced practitioners told them that emulators missed a lot of potential problems and the only way to be really sure is to test on actual devices.

From the bright light of that insight, Adobe Shadow was born. Simply put (this is a simple product), Shadow enables you to pair any number of mobile devices with your desktop machine and then browse the same page on all of the devices simultaneously. Any changes to the page made on the desktop machine are instantly reflected on the paired mobile devices. Now here’s the really cool thing (if you’re a web developer, at least), if something looks bad on one of the devices you can command-click on it and “inspect element,” just like on a desktop machine. And changes that you try out in inspector mode will be instantly reflected on the page on that device. The Shadow do! All of the guess work goes out of the process.

Shadow is being released as a free, open beta through Adobe Labs. The full, paid product release will come later in the year, but developers can get their hands on it right away. And Bowman assured me that the pricing will follow a two tier model that will make it affordable to smaller and larger shops alike.

As remarkable as this product will be for its users, what Shadow reveals about the transformations within Adobe are equally remarkable. Known for professional toolsets like Photoshop, InDesign and Acrobat—and for the avowedly anti-web-standard Flash—Adobe is moving beyond the purchased software model towards cloud based software as a service (SAAS). The subscription model provides a more consistent revenue stream and is more in keeping with the emergent consumer pattern of frequent downloads of inexpensive apps than the previous long-product-cycle model.

This shift in strategy frees Adobe up to pursue “minimum viable products,”
like Shadow, that really meet a specific need and get them out to users quickly—just like web startups do. This nimbleness is a neat trick for a large company.

Even more importantly, now that it has begun to decommission Flash in favor of web standard HTML5 solutions Adobe is now moving with the grain of the open web. This puts it in a much more neutral, agnostic—and helpful—position with the web development community, portions of which are vigorous open source partisans. Shadow is interesting because, unlike many previous Adobe products, this one work equally well with any code editor—not just their Dreamweaver or larger Creative Suite.

You can watch a demo of Shadow on the Adobe Labs site and download the components for free if you want to try it out. And if you are working with a developer on your site, make sure they’ve got it, because when it comes to device testing, why have a shadow of a doubt?

This article is available online at:
As a young designer, I had an audience with Alexander Liberman, the auteur/artiste who made Condé Nast an aesthetic powerhouse, who tried to convince me (unsucessfully) that editorial design was not very “creative.” What he meant was that, at the time, the balance of creative power in magazines was not on the side of the designer, particularly those of entry level. Fast forward almost 30 years and the picture is a little different. I met Scott Dadich last month at the Abstract Maine 2011 design conference in Portland, Maine. In many ways, the mantle from Liberman has passed to Dadich, who is now the vice president of digital magazine development for Condé Nast. Although far from running the show as his imperious Russian predecessor had, the team building Texan has already succeeded in a short time at placing design and creative direction at the center of what is new and important in magazine content production—and positioned Condé Nast at the forefront of magazine app publishers.

My old friend Evan Smith was the editor of Texas Monthly for most of the last decade, where Scott worked until he went to Wired in 2006. “Scott was always pushing, pushing, pushing. He always wanted us to do more, faster. For someone his age—he was in his mid-twenties at the time—he had an unusually mature sense of what he wanted to do and how he wanted to do it,” said Evan. “His command of the technology of design—the tools and the functionality—truly transformed us, truly made us a 21st century magazine. It was clear to me he had such high ambitions that we couldn’t contain him. And I was right. But his success hasn’t surprised me one bit. He’s more drive, personally and professionally, than any creative director I’ve ever known.”
Scott gave a great talk about how he jammed with a team from Wired and Adobe to reshape the magazine for the newly minted iPad. Much of the press about Wired’s app focused on the multimedia bells and whistles and attendant file size bloat. Most high-end magazine apps are open to this criticism, and most of them are trying to address it. But what’s worse: making your readers wait for a long download or having them be underwhelmed when the app opens up?

The most interesting thing to me, though, was how Scott and his team came up with a new architecture that specifically takes advantage of the tablet’s gestural potentials. Digital magazines of the page flipping pdf sort have failed to capture the imaginations of consumers. When I said something to this effect in a previous post, Marcus Grimm, the marketing director of Nxtbook Media, commented (correctly) that digital magazines “have very much come into their own in the BtoB space.” He should know, Nxtbook produces a ton of them and has been in the game for almost a decade. But the fact is, a lot of those magazines no longer exist in print because the economics don’t make sense. So the BtoB audiences are engaged with these digital editions because they need the content and this is the way to get it.

As Dadich well knows, consumers have lots of choices and many have not been choosing print magazines. So one of the big choices these new tablet apps give consumers is to flip less pages. When they want to skim the top pages they flip through and when they want to “drill down” or “dig deep” they do just that—with their index fingers. Here’s the conversation I had with Scott recently about how these new shapes stacked up:

The whole idea of the stack—of the vertical and horizontal navigation...how did that emerge?

In late summer/early fall of 2009 we had decided to partner with Adobe to make the tablet magazine idea a reality. It was me, Jeremy Clark who is the director of the XD (Experience Design) group at Adobe, and his team consisted of Bruce Bell, Justin Van Slembrouck and Amy Haynes. So the four of them and then me and Margaret Swart, Managing Art Director and Wyatt Mitchell, Design Director, from Wired.

So the 7 of us really acted as a team for 6 months, working on all of these ideas. One of the things we did that fall was to do a content audit of 4 issues of the magazine where we broke apart all of the articles by section, by article length, by article type. Just as an exercise we oriented them in what are now stacks, to look at article lengths and design fidelity and the kinds of moves and investments we were making for those stories. So it’s like hanging everything on a clothesline—you can really quickly see the differences between all those different pieces.
There was a time there where we were thinking of this idea that you would swipe right/left to engage an article and then you would swipe up or down to get to the next article. Bruce and Justin felt pretty strongly that the metaphor of the clothes line was really the right way to go. So we tested both and really fell in love with what we were calling the clothes line at that time and it just seemed really, really natural. We had a convertible Dell touch screen laptop. Bruce and Justin had built a little AIR app that would let us do that—engage that browser metaphor—and so we started playing with it and really liked it. So I think we were using it around the office and I coined the word “stacks” for a stack of content—Apple uses the term stacks in their spring-loaded docking item—so it sort of looked like that and so I just used the word “stack” because we were having trouble differentiating internally at Wired between digital and print. We had internal clarity almost to a fault! We kept saying digital story, a story, pages, digital story, pages and it was just too cumbersome. So we bifurcated and from then on used the term “story” for print and “stack” for digital and “pages” for print and “screens” for digital. An 8-page story could be a 16-screen stack. From then on everybody got it. It was very quick and easy to talk about how we were structuring content in the digital edition.

*When Gael went to start working on the Martha Stewart stuff, was she working with the same team from Adobe?*

Yes, I showed Stephen (Doyle) and Gael my prototype back in the Spring of 2009 before I had even started working with Adobe and at that point the prototype didn’t have the stack metaphor in it. It was only when I started working with the Adobe folks that we settled on that. So she was familiar with the work I was doing and had wanted to engage on a digital edition of her own, so I introduced her to the Adobe guys and she started working with Bruce and Justin and Jeremy as well as Kenji Arakawa, another member of the team that was based here in New York.

*In terms of your own personal survey of digital magazines at this point, are there a lot of people who’ve come to the same conclusion? Either through simultaneous invention or because they saw your thing and thought, “we can do that, too”.*
I think it’s both. The fundamental import from stacks was a combinative effect—like I talked about at the conference—like reading and scrolling on an iPhone is performed on the vertical and typically magazine browsing is that single axis of right/left movement so it was intended to always be that sort of hybrid. So, right as we were working on that notion, the MAG+ guys—in Sweden—had put up a video and they had described almost the identical notion. They weren’t calling them stacks but it was the same right/left, up/down experience. I think the initial Time Magazine had some factor of that—now they’ve flattened out and they’re just doing single access. But I see more and more of it and especially as the Adobe platform has been adopted there are two or three hundred publishers on that. And then the things like the FT(Financial Times) web app that came out recently and uses the same metaphor. So I think it’s starting to gain some traction, which is really gratifying.

And you’re seeing it both with people who are using Adobe and people who are doing from with other platforms?

Both Woodwing and Adobe enable the metaphors so it’s not really about a technology choice it’s more of an architecture choice.

Yeah. Adobe’s real advantage is that they’ve already got the majority of publication designers using their software.

Exactly and that’s how we’ve been able to roll out across nine titles in monthly publications and we’re going to be up to 12 here before long in non-regular. Vogue is producing from time to time and there are a couple of other titles here that are going to come out with some things shortly. So it’s not foreign to the team. It’s the tools that they have on their machine. It’s literally hours worth of training. It’s more about the whole systems work—that’s actually the real hurdle. Content creation is actually the easiest part of it. It’s time management and work flow, the commitment to the manufacturing of the digital edition. That is actually the bigger hurdle to a magazine team.

There’s another interview with Scott by Justin Ellis on the Nieman Journalism Lab, and look for part two of this interview for more on appazine on-ramps soon.