Scripps Network

Content Management System Design

Design Research, Vision, User Requirements

Version: 1.6

Date: 11/05/07

Sean Patrick Coon

Principal Designer, dotmatrix studios

.matrix

Table of Contents

Problem Statement **3**
- Paradigm Shift 3
- Usability Issues 4

Vision Statement **5**
- Role-based user requirements 5
- Simplify and Reduce; Create an Elegant System 5
- Enable Discovery and Dynamic Relationships 6
- Decouple Code From Content From Styling 6
- The Holistic Vision 7

User Research **8**

Personae Expectations **9**
- Primary Personas 9
- Secondary Personas 17

Context Scenarios: Bridge to the Interface **19**

Key Path Scenarios: Mapping Form and Function **40**

Problem Statement

Until only recently, Scripps Network's business model was steeped in media and newspapers investments with an internal focus on supporting such platforms. The current content management system is a legacy investment of the newspaper dominated years.

Since the company has awakened to the potential of the Internet, a shift in focus has occurred and the Scripps online brands are now looking to catch up to the Web 2.0 revolution.

Paradigm Shift

In the real world, editors must make explicit calls regarding the placement of content within magazines and newspapers pages, with little insight into the performance of content sections and subsections and their readerships shifting focus of attention.

The Internet disrupts the norms of real space. We're no longer visiting destination content sites online as we do offline with particular publications or as we did in the previous pass of the first pass of the Internet. We're skipping the layers of man-made organization found in brand.

Now that we're knee deep in the Information Age, Google search results drive the vast majority of page traffic. Advancements in information retrieval have made it possible for us to receive relevant matches to natural language queries. This allows us to tap into the long tail of online documents, returning relevant information objects regardless of the marketing budgets behind a particular domain.

FastForward was designed in an age that didn't compete with or need to leverage Google PageRank. In 2001, there weren't millions upon millions of bloggers busy creating content, steeped with meta-data and creating endless options for participating in exterior tag namespaces.

With the advent of folksonomies, information architecture approaches—such as controlled vocabularies, which drive contextual retrieval within organizations based on the application of explicit meta-data—have been opened up to the world to construct. The average person can now participate in the organization of information; on a micro level (their blogs) a macro level (del.icio.us) and domains found in-between (Amazon.com).

And people are participating more and more each year.

This global participation not only affects individual brands by reducing the desire for surfing destination sites. Google is now the top “destination site” as it were. This increase of publishing participation and organization has reduced the value of landing pages within top-down classification systems of all domains, as people are more apt to search these days instead of browse.

While shallow topical navigation will always establish domain focus and brand position, drill-down navigation is now primarily used as a navigational last resort these days. Search and contextual links to relevant content are much more intuitive and efficient discovery devices..

The good news is that with six highly established media brands driving traffic, Scripps Networks is in a unique position to capitalize on destination readership and unique search visitors.

Usability Issues

Since FastForward was originally designed to serve the publishing needs of a newspaper, the current system is full of dead features that cause cognitive friction for users.

Issues that affect internal productivity and content quality include:

- Poor publishing preview features : designers and editors cannot view content as it would be displayed in a production environment, which contributes to wasted time and poor content

- There are no previews for images : can triple the time and effort to locate the right images

- Limited search functionality : search is not taking advantage of the meta-data found within assets, but even if it could current asset types lack quality structured meta-data that should be assigned at upload

- Link lists, images, content blurbs, etc. are hard-coded : the HTML of landing page modules requires content engineers to update content, which is an investment in a team of three FTEs

There are many more usability issues with the system – too many to list in this high-level document. Each of these issues impacts productivity, the bottom-line of the business and the potential for creating a world-class user experience.

Vision Statement

The biggest mistake with a new CMS would be for us to attempt to grant every feature wish out of the gate. If we head down that path we'll end up with a system that serves everyone, yet no one in particular, and the cost and time to build will be unmanageable.

Role-based user requirements

To design a content management system that meets the explicit needs of the people that use the system, we're interviewing every person that currently touches the CMS and *who might touch it* in an optimal production and publishing process.

This research will help us create proposed usage scenarios based on archetypical design persona needs, and focus the first release of the redesign on role-based key path scenarios.

As we move forward, subsequent releases will take to be determined edge-case scenarios into consideration, as well as new input from future interviews.

The design personae and scenario libraries will evolve over time with additional input from interviews and feature requests, mirroring the actual goals, needs and desires of system operators while supporting the direction and objectives of the business.

The following are design principles we intend to pursue.

Simplify and Reduce: Create an Elegant System

The essential mechanisms for gathering assets, producing content, assigning attributes, indexing articles, etc. are in place in the current system. The problem is that the underlying system is dumb by modern standards on many fronts, and the presentation layer is practically unusable in numerous instances.

As a general design approach, we plan on eliminating cognitive friction from the interface—whether that means streamlining functionality to improve focus or overhauling the complete interaction model within task flows.

The approach will be to focus on key functionality and paths, and then simplifying the user interface, making the system both usable and useful. Everything else is noise.

The purpose of this project is to design a CMS that meets the explicit needs of explicitly understood and modeled users. Subtraction may not equate with the

complete removal of features; it could very well mean that one permission type has access to a particular set of features, while another does not.

Enable Discovery and Dynamic Relationships

The average blogger with a basic Wordpress or Blogger setup currently has better tools for creating meta-data hooks in their content than Scripps editors do with FastForward.

Once these tags are applied to their posts, content is instantly indexed within their own domain and across the web in any service using the same protocol. This greatly increases the potential for findability and increased PageRank. This simple ability to assign concept attributes to information objects and expose them in the post interface is one of the reasons that blogging has become such a powerful platform for increasing relational discovery.

Our new CMS must empower editors with the ability to increase the quality of article meta-data. The greater the degree of structured tagging applied to the article level, the better the chance that dynamically generated modules of related articles will truly relate to the host article.

Once articles start being tagged, the additional benefit of exposing cross-brand relationships will become available. This will allow Scripps to generate relational discovery across their multitude of web properties, increasing visitors and advertising revenue.

Decouple Code From Content And Styling

The current publishing workflow has been built around the inadequacies of the limitations of the current CMS, not the needs of the business and content teams.

We propose decoupling all presentation code from managed content and styling options. This will shift the module creation process from code snippets living in the system to modules with admin screens, allowing brand managers to quickly drop into any top-level modules and *schedule changes*.

This one shift will enable content engineers to focus on more important elements of the publishing process and empower brand teams to become much more agile. This will also change the dynamics for sketching new interfaces, as there will be an actual network-wide library of module design patterns to choose from.

Product managers will have a much better sense of short-term options available to them and designers will be able to quickly customize modules to meet brand requirements instead of always starting from scratch.

The Holistic Vision

These are our initial set of design principles driving our approach to the new content management system. They will stand on their own as newly discovered user scenarios are added and clearly mark fundamental differences between the current system and a future state.

By creating such a system, the brand, advertising, design, development, SEO, search and editorial teams will be in a position to be successful in delivering world class, targeted and engaging content while increasing viewership and ad revenue.

User Research

Design Personas are modeled after running individual, one-on-one interviews or contextual inquiry with representative users of a particular system or domain. This process simultaneously qualifies and rejects stereotypical assumptions of user goals and tasks, in turn producing a narrative, forward-looking, archetypical understanding of individual users of the proposed system.

The following design personas were modeled after observing and interviewing five designers, five editors, two content engineers, six core brand teams, two network sr. vice-presidents, the ad sales group, the search team and the SEO team.

Once the primary set of user personas have been qualified by the client, the next step will be to flesh out context scenarios for each persona. This will be considered the final deliverable of the Discovery project and the input for the Planning Phase.

Personae Expectations

Primary Personas

EDITOR, ANGELA DAVIES
Focus: Publish articles, index/tag articles, manage assets, SEO

PRODUCT MANAGER, JEFF POLINSKY
Focus: Design templates, manage assets, schedule/update content

CONTENT ENGINEER, SCOTT McBRIDE
Focus: Template Creation and Management

AD SALES, BARBARA JONES
Focus: Sponsorship Management

SEO, DAVID MIMS
Focus: SEO application, Manage assets

DESIGNER, SARAH HUAN
Focus: Design modules & templates, Manage assets

LIBRARY SCIENCE, JENNIFER STEWART
Focus: Manage metadata & taxonomy

PRIMARY DESIGN PERSONA

ANGELA DAVIES

EDITOR

"I LOVE EDITING STORIES FOR A LIVING. PLEASE DON'T LET THIS DARN PROGRAM MAKE ME HATE MY JOB"



27 years-old
Editor

DESIGN MODULES/TEMPLATES

IMPLEMENTATION

PUBLISH ARTICLES

INDEX/TAG ARTICLES

MANAGE ASSETS

SEO APPLICATION

SCHEDULE/UPDATE CONTENT

SPONSORSHIP MANAGEMENT

As an Editor for one of Scripps Networks most popular brands, Angela's responsibilities range from writing original content to editing and publishing other author's work. She rather enjoys her work, taking immense pride in the final product, as each published article gives her a real sense of accomplishment.

Because of her role in the publishing process, Angela has an intimate understanding of the brands she serves and their unique classification systems. Each article is to be explicitly assigned to the brand taxonomy and it is Angela's responsibility alone to assign such relationships. The problem is that indexing articles in the current system is both a shot in the dark and a chore, as the taxonomy is presented in a flat list with cryptic lingo. Angela does understand the importance of creating article specific meta-data, but the current system doesn't give her a sense of the tags she's applied over time, so it's hard to be consistent with terminology.

Article pages within the CMS are chock full of superfluous features left over from the newspaper business, which not only makes publishing more time consuming, but obfuscates the need for features that support publishing content to such a dynamic and relational place as the web. Case in point: When Angela wants to arrange the layout of an article, she's forced to abstract each element – text, images, even links – into individual, manageable components and arrange the article as if she were perched on a mountaintop, 15,000 feet up in the air.

Angela looks at the mechanics of her job as being rather simplistic – produce content, collect assets, assign links, massage layout, index/tag the article and publish – but her days are never as simplistic as they should be.

Primary goals/needs

- A simplified article-level publishing interface – focusing on layout, indexing and tagging.
- More powerful article layout features, such as pagination, modular slide shows, rich text editor, etc.
- Dynamic indexing and tagging interfaces – auto-exposing related branches (taxonomy/packages) and suggested tags
- Cut down on time finding/managing images

.matrix

PRIMARY DESIGN PERSONA

JEFF POLINSKY

PRODUCT MANAGER

"LET'S GET BEYOND STICKINESS AND SHOOT FOR COMMUNITY"



43 years old
Product Manager

DESIGN MODULES/TEMPLATES

IMPLEMENTATION

PUBLISH ARTICLES

INDEX/TAG ARTICLES

MANAGE ASSETS

SEO APPLICATION

SCHEDULE/UPDATE CONTENT

SPONSORSHIP MANAGEMENT

As the Product Manager for one of the largest online brands in Scripps Networks, Jeff's short-term tactical approach and long-term vision is primarily responsible for its success moving forward.

With on-air creating such a huge traffic flow to the site, Jeff could play it safe by simply refreshing landing pages, continuing to sell sponsorship packages and riding the wave of new content generated by his editors each day. He could, but he won't; Jeff knows that the web is evolving daily and he wants his brand to be on the forefront of innovation, rather than the tail end.

Traditionally, traffic has been the greatest indicator of web success. While advertising sales won't be moving away from that metric for a while, new indicators of domain valuation are beginning to make themselves apparent. The rise of community around participatory reviews and content submissions is turning traditional programming on its head. While these new elements are catalysts for producing additional traffic, they're generating something even more valuable: loyalty.

In the short-term, Jeff needs to be able to elicit change on the site much quicker. He can't wait for resources to be available to make the simplest of tweaks. Jeff wants to cut out the middleman scheduling content changes – if not by himself, then by a member of his core team. He also wants to be able to tap into the functionality created by other brands. No use in reinventing the wheel.

Long-term, Jeff is looking to open up the site to incorporate dynamic, community-centered functionality. He's not exactly sure what that means yet, but he's already playing around with social networks. Give him six months and he's bound to have a solid approach mapped out.

Primary goals/needs

- To be able to update top-level page modules with new content, links, images, etc. in the blink of an eye
- The ability to reuse functionality from across the network and make it brand specific with relative ease
- To have access to dynamic methods of presenting relevant information

.matrix

PRIMARY DESIGN PERSONA

SCOTT McBRIDE

CLIENT-SIDE DEVELOPER

"THE PRESENTATION LAYER ISN'T CHILD'S PLAY"



33 years-old
Client-side developer

DESIGN MODULES/TEMPLATES

IMPLEMENTATION

PUBLISH ARTICLES

INDEX/TAG ARTICLES

MANAGE ASSETS

SEO APPLICATION

SCHEDULE/UPDATE CONTENT

SPONSORSHIP MANAGEMENT

Scott has been developing front-end code for the past 12 years, beginning his career as a Lingo scripter for multimedia games. Since he moved explicitly to the web, he's dealt with every browser challenge imaginable – from weak JavaScript support in early iterations of Netscape to limited uptake of the Flash plug-in as broadband slowly ramped up across the country. Before AJAX was coined, Scott had created workarounds to provide similar asynchronous functionality for his designer counterparts. Throughout his career, his knowledge has only been superseded by his ingenuity.

As the User Experience team becomes specialized with Designers focusing on best practice design patterns instead of scripting the presentation of their designs, Scott increasingly finds his hands full. Projects can range from scripting the behavior of simplistic content blocks – lists, articles & images, slide shows, Flash promotions, etc. – to fully functional applications, such as the Recipe Box.

While his approach to quality remains consistent – the CSD team is committed to producing standards compliant code that works across all platforms and browsers – Scott's process is driven by the needs of each brand. Certain projects call not only for the presentation to the end user, but also for admin screens that internal employees use to generate new versions of content presentation (blocks). These screens aren't produced with the same cross-platform and browser standards, but they are created with the same eye for detail, ensuring solid usability.

Moving forward, Scott wants to make sure that his group isn't left out of newly formed workflows. It's extremely important that they have access to a staging server and to test all code in a replica state of production. Scott's excited by the possibilities of the new CMS… as long as his needs are explicitly taken into account.

Primary goals/needs

- To have direct access to a staging server for QA'ing his work
- To create content admin screens with as little effort as possible
- To have explicit approval in the workflow before styled assets or configured templates go live.

.matrix

PRIMARY DESIGN PERSONA

BARBARA JONES

AD SALES

"WE NEED TO SUPPORT OUR CURRENT BUSINESS MODEL, FIRST AND FOREMOST"



46 years-old
Ad Sales

DESIGN MODULES/TEMPLATES

IMPLEMENTATION

PUBLISH ARTICLES

INDEX/TAG ARTICLES

MANAGE ASSETS

SEO APPLICATION

SCHEDULE/UPDATE CONTENT

SPONSORSHIP MANAGEMENT

The Ad Sales group at Scripps Networks plays a key role in shaping the user experience for the network – not as much with the editorial approach of each brand, but very much so in how individual templates end up being structured to optimize ad impressions.

Barbara has been working on solutions to better incorporate sponsorships into the context of pages – providing a more relevant experience for both the end user and the advertiser. It's her belief that useful functionality – related to the content and contextual to user needs on a particular page – is a better vehicle for sponsorship than traditional ad inventory (i.e. big boxes, banners, etc.), as the user will attribute usefulness to the sponsors themselves.

Ideas like these are floating between her team and the brand product managers, but Barbara's immediate responsibility is to deal with the tactical realities of the current Ad Sales initiatives, which includes creating and selling packaged content to individual companies.

Barbara and her team have created a particular valuation for home pages and section fronts. While their sales process is static – sales people work off a spreadsheet of available inventory that is only updated once per week – they are still able to effectively communicate closed deals and generate new sponsorship codes to be assigned to various levels of content.

The major issue Barbara runs into with sponsorship packages is that content indexed to the package index does not retain the sponsorship code of the overall package. A simple mechanism to conditionally present a package level ad and related links module on lower-level content when the user directly navigates to the article level content from the package home page would be huge.

Primary goals/needs

- To be able to directly assign sponsorship codes to content or packages
- "Smart sponsorship" – so advertising follows the user down from a package page to an article and across siblings
- Discover dynamic ways of bundling and auctioning sponsorship opportunities

.matrix

PRIMARY DESIGN PERSONA

DAVID MIMS

SEARCH ENGINE OPTIMIZATION

"LANGUAGE IS WHAT SEPARATES US FROM EACH OTHER"



39 years-old
SEO Manager

DESIGN MODULES/TEMPLATES

IMPLEMENTATION

PUBLISH ARTICLES

INDEX/TAG ARTICLES

MANAGE ASSETS

SEO APPLICATION

SCHEDULE/UPDATE CONTENT

SPONSORSHIP MANAGEMENT

David's job is to ensure that the Scripps Networks' online properties have the greatest chance for discovery visa vie people using modern day search engines. In order for this to occur successfully, David and his colleagues approach their jobs from a few different angles:

System driven page structure

- Title tag output of [Article title] > [Section] > [Brand]
- URLs with keywords & dashes in-between
- Every article has only one permalink
- Utilize redirects from old website to new that can be followed by search engines - 301 and 302 redirects. Redirect traffic to specific pages and not all to homepage, if possible.
- Integrate SEO title and description tags into search results

Guidelines for editors

- Fields to give editors the option to create unique title and meta-description on multi-page articles
- Alt-tags on all images

If the editors are consistent in their approach and the system has been designed to take all template optimization into account, David's responsibilities have been narrowed down to R&D and QA. R&D happens each time a major search engine tweaks their algorithm – David and team come up with a plan for taking the evolution into account – while QA occurs on an ongoing basis.

When home pages, section fronts and packages are ready to hit production, it would be helpful if SEO had the ability to bless the optimization prior to launch. Conversely, SEO would only want access to a feed of article data points as they stream to production, as there's no reason to become a bottleneck in publishing. If there are issues, they should be able to edit SEO components as necessary.

Primary goals/needs

- To have the system take into account all modern SEO approaches when building articles and pages
- To have access in/outside of the publishing workflow to QA SEO application to pages and articles, respectively.
- To be able to apply search optimized keywords to editor's tags

.matrix

PRIMARY DESIGN PERSONA

SARAH LUAN

WEB DESIGNER / NETWORK

"A BETTER USER EXPERIENCE MAKES A BETTER BRAND"



32 years-old
Sr. Web Designer
Network

DESIGN MODULES/TEMPLATES

IMPLEMENTATION

PUBLISH ARTICLES

INDEX/TAG ARTICLES

MANAGE ASSETS

SEO APPLICATION

SCHEDULE/UPDATE CONTENT

SPONSORSHIP MANAGEMENT

As a Designer for one of Scripps' top trafficked web properties, Sarah's job is to iteratively improve the user experience of the site. That broad description includes everything from designing better widgets for increased findability to managing freelancers designing package sections to collaborating with the network design team on establishing design patterns across brands.

Prior to client-side development being built out as a support team, Sarah was knee deep in the system, spending a large percentage of her time scripting code to support her own designs. Now only occasionally is Sarah called upon to assist with updating the styling of section front modules, so her interaction with the CMS has been reduced immeasurably.

A vast majority of Sarah's work today is based around forming modules and page layouts as new design patterns in the Scripps Network user experience library of functionality. While her particular brand poses unique challenges that call for specific design solutions, Sarah's overall creative process is steeped in customizing existing network patterns to fit her brand experience.

To a designer with less experience, such a design process might feel limiting, but Sarah has enough experience to understand that pure "creativity" at the cost of interoperability, best practice usability and consistency is not a sustainable approach – not only for the end user experience, but the business itself.

Primary goals/needs

- To be able to customize style elements of pre-existing modules for her brand experience *without* digging into code
- To reuse modules from across the network in new page designs
- To be alerted when new modules have been made available in the library

.matrix

PRIMARY DESIGN PERSONA

JENNIFER STEWART

LIBRARY SCIENCE

"YOU FOCUS ON THE CONTENT; I"LL GET META ABOUT IT"



29 years-old
Library Science

DESIGN MODULES/TEMPLATES

IMPLEMENTATION

PUBLISH ARTICLES

INDEX/TAG ARTICLES

MANAGE ASSETS

SEO APPLICATION

SCHEDULE/UPDATE CONTENT

SPONSORSHIP MANAGEMENT

Jennifer works as a library scientist in Scripps' Media Center, maintaining the ever-evolving domain taxonomy and metadata associated with each Scripps Networks online brand.

On any given day, Jennifer's job can take her knee deep into the depths of the most granular assets of the domain – metadata – or keep her occupied at a macro level, dealing with brand taxonomies and strategic initiatives to harness the power of search as a revenue generator.

As an academically trained LIS, Jennifer has a unique perspective of "organization." She implicitly understands that traditional classification systems have become marginalized overnight when introduced into the relational construct and ecosystem of the World Wide Web.

There's no such thing as "spaces" online; no absolute limits to describe objects as there are in the real world. By slowly accepting and understanding this basic notion since graduation eight years ago, Jennifer has come to embrace the balance between structured data and messy metadata. The former serves the master of brand position, while the latter serves the master of findability and discovery.

Jennifer interacts with numerous groups within Scripps Networks, but her primary function is to assist Editorial and SEO as they attempt to do their jobs adding metadata into the domain. She doesn't interfere with them doing their jobs, but she does have the authority to tweak their metadata input based on domain standards developed by her team. Jennifer needs a simple way to validate particular instances of metadata when they enter the system without burdening the groups responsible for the additions.

Primary goals/needs

- Curate the corporate taxonomy as a living entity
- Validate new facets and tags as they are proposed
- Authorize changes in popular lexicon
- Expand and contract facets of taxonomy as needed

.matrix

Secondary Personas

FREELANCE DESIGNER, JASON FRENCH
Focus: Design modules & templates, Manage assets

FREELANCE WRITER, TINA GORDON
Focus: Publish articles

FREELANCE CONTENT ENGINEER, CAMERON JACKSON
Focus: Template Creation and Management

SECONDARY DESIGN PERSONAE

FREELANCE STAFF

PUBLISHING PROCESS



Jason French, Designer

As a freelance designer for the UX Team, Jason works on a project-by-project basis, usually designing sponsorship packages.

Since the team is splitting design into UI and CSD, Jason's role will probably be reduced to specifying the presentation of package pages. As such, Jason will need simplified access to existing design templates and modules, so he can peruse and reuse their functionality as much as possible.



Tina Gordon, Writer

As a freelance writer, Tina's work appears in a handful of weekly publications, primarily Do It Yourself type magazines. She'll receive an assignment from the editorial staff at Scripps and return a Word document, ready for an Editor review and approval.

Tina does not have the responsibility of indexing, tagging or publishing the articles she writes; she simply needs to be able to drop her work into an article form and save it for Editor review.



Cameron Jackson, CSD

Cameron is a hardcore client-side developer, serving numerous clients out of his home office. His projects with Scripps are often soup to nuts, where he needs to communicate not only with the design team (internal or freelance), but both the client-side and software development teams as well.

When it comes to interfacing with the new CMS, Cameron needs to be able to test his code on a staging server and generate admin screens whenever necessary.

Context Scenarios: Bridge to the Interface

Context scenarios describe the broad context in which usage patterns are exhibited and include environmental and organizational considerations. These scenarios establish the primary touch-points that each primary and secondary design persona will have with the system and potentially other design personae over the course of a day, week or other period of time that provides context to goal directed behavior.

A context scenario will not represent system or workflow behaviors as they currently exist. This exercise should read like a black box of scope and design opportunities.



ANGELA DAVIES
Publishing, Context Scenario

1. Angela has begun to check her cell for work updates on the way to work. She usually waits until she gets into the office before worrying about potential last minute work, but the way the last week has gone – two new shows to support and an extra freelancer to manage – she doesn't want to take the chance being blindsided. Her SMS inbox is empty, so her queue feed hasn't changed since yesterday. Phew!

2. After grabbing a cup of coffee, Angela makes the morning rounds with her fellow editors to see if anything pressing might be on the horizon. So far so good. Once she settles into her cube, she logs into her CMS to get started on her day. As far as Angela knows, she has three articles to work on today.

3. Upon logging in, Angela's dashboard clearly presents her publishing schedule, which includes her three articles due by the end of tomorrow, a queue of work-in-progress drafts (including articles entered and saved by her freelance writers) and a list of the most trafficked and searches articles across numerous brands and taxonomies. She saved her DIY article in a draft state yesterday, so with a click on the title name at the top of her draft queue she's now back into her work.

4. In order to write this particular article, Angela needs to watch the episode, take notes and find key screenshots that help describe the Howto, step-by-step nature of the show. Angela logs into the Scripps broadcast video service that houses compressed versions of all on-air shows. As she watches this particular episode, Angela pauses the playback on key frames to mark them as potential screenshots for use in her article. Each time she marks a frame, she also takes a moment to give it an explicit description, useful later on as both an alt tag and a caption. Once finished marking frames, Angela focuses on writing the article, knowing that higher-quality grabs will be made available to her once the system processes her requests.

5. Angela continues to construct the rest of the article. Once the narrative is complete, she goes back for a style pass using the rich-text editor to emphasize words, create bullet-point lists where necessary, etc.

6. Once the concept of metadata and tagging was explained to Angela, she completely understood its value to the organization. On this particular article type, there are four facets of tag classes; Angela tags the article for Material, People and Room, finishing it off with a free form area for descriptors.

7. As she is tagging the article, Angela notices that the talent of the show isn't represented in the controlled vocabulary of the "People" tag class, so she uses an additional field to add the new name to the facet. Those library science folks are going to love getting that notification.

8. As her tags are being entered, image thumbnails with particular bits of metadata automatically present themselves in the interface. Angela is looking to illustrate this article even more with an image of a particular deck, so with a few clicks she narrows down which tags are being used in the matching pull. The set of image thumbnails respond to her tweaking by instantly reducing the choices to a wide-range of deck sizes, shapes, locations and materials. One more click on the material tag in the article and the image set reduces to four images of oak decks. Angela chooses one of the two images that show up with a tag as never been used on DIY before and places the image in the proper context of the article.

9. As Angela finishes finding and adding her original image to the article, she's alerted that her episode screenshots are available and ready to use. She associates the images to her article and saves it. After a quick preview to make sure that it's rendering correctly on the staging server, Angela publishes the episode, which automatically sends a notification to her manager, David Mims in SEO and Jeff Polinsky, the Product Manager.



JEFF POLINSKY
Ideation & Sketching, Context Scenario

1. Jeff is fixated on constantly improving the user experience of his brand, so he works rather closely with Sarah Luan to brainstorm new approaches for improving both the usability and usefulness of his site.

2. Last week, Jeff found himself exploring one of the network's sister brand homepages and found an interesting module. It presented the top 10 articles by views and the number of times emailed. As it currently stands, Jeff's home page is clean and well structured for Ad Sales, but it doesn't present content options in a manner that reflects the zeitgeist of his users.

3. Jeff immediately logs into the CMS and heads directly for the template & module library. He browses the modules attributed to the other network brand and immediately finds the module in question.

4. Once engaged with the module, Jeff is presented with all of the available, baked-in data presentation options. Not only can he set it to display top 10 articles by visits and emailed, he can also choose to display most linked-to and most commented. Jeff adds the most linked-to option and decides to change the output setting to top 5 across the board. Jeff saves this new instance and gives it a description.

5. Once saved, Jeff clicks the preview link to see how it behaves. Not only has it generated the correct data display, but also the system was smart enough to swap out the old brand's style sheet for Jeff's brand. Sarah might want to tweak particular elements of the styling, but for now it completely fits the style of his current homepage.

6. Jeff moves across the library to his theme section and finds the home page template. He spends about twenty minutes sketching up a new home page by

moving modules around on the screen in various positions, adding new ones and editing their behaviors. Some modules seem to have built-in warnings that present themselves when Jeff places them in particular areas on the screen. It doesn't stop Jeff from arranging the template as he sees fit. He saves a new version of the home page with the new module he just created, though it's not going live just yet; Jeff needs approval from his designer before he can move this new design into Scott's CSD project queue. As soon as the template is saved, Sarah is pinged that Jeff is playing around again.

7. Once Sarah and Jeff review what he's trying to accomplish, they both realize that he needs a few other modules with particular behavior for the home page. After searching the module library to no avail, Jeff comes to the conclusion that they need to create these new modules from scratch.



JEFF POLINSKY
Updating & Scheduling Content, Context Scenario

1. Jeff balances the top-level pages of his brand with dynamically generated content (i.e. article lists) and modules that are crafted by a human voice. This is what seems to drive deeper readership. Jeff and the brand team have devised a twice weekly schedule for updating content, images and links on top-level pages across the brand.

2. Every Wednesday and Friday, Jeff reviews page performance, drops into the live theme library in the CMS, and changes pictures, replaces links and writes new headlines for any number of modules. Some days he publishes the changes instantly; some days he sets the publishing time for a day or two away. In any case, Jeff doesn't need to wait for a resource to free up to make simple alterations.

3. This month the team released a useful feature update; the site evolves to populate modules with content that matches what's popular across both the suite of brands that Scripps offers, as well as the zeitgeist across the internet.

4. Jeff had his doubts at first, but his concerns around "getting it right" proved to be negligible, as the content available is all above board and the analytics have proved that automation more often than not increases readership, which positively impact ad revenue.

5. While he still wants to tweak the content and arrangement of modules, Jeff receives analytic updates via weekly emails, which expose readership changes, including projected impact on ad revenue. Jeff is more than happy to let the system do the heavy lifting, and now has the focus to invest his time elsewhere on new revenue streams.



JEFF POLINSKY
New Module Creation, Context Scenario

1. After reviewing the CMS module library from top to bottom, Jeff realizes that he can't leverage any existing modules and that he needs to create two new types for his latest homepage evolution.

2. Jeff opens the module creation interface and chooses the feeds he wants, selects new display attributes, and the featured articles and videos. He previews his work across multiple display formats and adds the modules to his library.

3. As Jeff adds the module, Sarah is pinged to cosign that the new module conforms to the design system that she and her team devised for each platform. This shouldn't be an issue, as the engineering team build limitations into the module creation app that conforms with the platform pattern library. Sarah appreciates the visibility into the organically evolving system and willingly provides quality control as necessary.



SCOTT McBRIDE
New Block Creation, Context Scenario

1. Since the design team began focusing on the creation of design patterns for responsive modules, Scott's job has become much more manageable. Now when he's called to create new functionality for a particular brand, Scott knows that his code will be reused at some point, which greatly reduces his time spent on reinventing the wheel.

2. Once Sarah finishes detailing the behavioral and visual specs for new patterns, Scott begins working on the presentation code. He normally taps into the CMS' library of data calls to the backend, but with this particular functionality, he needs to wait for software development to generate APIs for accessing Omniture and Dart data. Once development adds the new hooks into the development library, Scott ties together the presentation and the backend calls.

3. Scott pings Jeff and Sarah to review the presentation behavior of the block, Scott enters the code into the CMS so that it's findable in the module library when Jeff and/or Sarah sketch out its use in a template.

4. Based on the content and dynamic data, Scott also creates admin screens to allow Jeff to update the module moving forward. While Jeff likes the fact that it'll be easy to edit the content, Scott loves that this extra step will reduce his workload down the road when another product manager or designer leverages the module for a different brand.

5. Finally, Scott creates an admin interface with explicit fields that alter the styling of the block while communicating the changes back to brand specific CSS definitions.



SCOTT McBRIDE
Producing Final Template, Context Scenario

1. Scott is in the midst of a number of projects when he logs into the CMS and finds an alert from Jeff Polinsky. It looks like Jeff has begun sketching a new home page from the various blocks available in the CMS library. It doesn't comply with the current design system, which is a feature, not a bug.

2. Once the layout is given a greenlight by Sarah as a usable design, Scott stitches together the necessary code to make these changes possible. The time saved by product management sketching their needs frees Sarah up to focus on new interfaces, and empowers Scott to quickly develop responsive code.

3. Scott reviews the final code in the staging server and shares out to the team. Once he gets the greenlight in the CMS, he checks in his code, making the template available to Jeff to manage.



BARBARA JONES
Discover, Create, Sell and Produce Dynamic Package, Context Scenario

1. Barbara knows that there are endless sales opportunities to be found within the flood of metadata generated by the Editorial and Broadband teams on a daily basis. Each day she steps in the office, she's confident that she will uncover hidden value.

2. As soon as she logs into the CMS, Barbara is presented with a view of inventory packages deemed most valuable based on metadata combinations created by an algorithm that takes criteria like season, events, traffic and coverage into account. Barbara has the opportunity to create her own packages by assigning metadata combinations, but the algorithm has generated great results on its own.

3. Thanksgiving is coming up around the corner, so Barbara is alerted to a metadata combination value proposition that uses the following primary tags; Thanksgiving, turkey, football, guests and entertain. Barbara clicks on the suggestion and views a results screen that presents various slices of articles, photographs and video matching the metadata being used. It also exposes expected revenue, including the diff from similar packages this time of year historically.

4. Barbara likes the presentation itself, but in this instance, she wants the page to only pull from FN, FL and HGTV. While narrowing the brand and taxonomy parameters to fit her needs, she also sets FN as the primary content provider. The page redraws and presents a new universe of content results and revenue projections.

5. On top of dynamic inventory, Barbara knows of three past sponsors that would engage with this kind of package. She assigns the potential sponsors to the package, creates three different sponsorship campaigns and previews the various campaign layouts in the presentation. Satisfied with her efforts, she

locks in the three potential sponsors and the system reaches out to their media-buy teams, which include a link to preview her mock campaign layouts within the context of the dynamic package.

6. The following day, one of the potential sponsors responds to her message with the intent to sponsor. Bob Jones of Fresh Market wants to sponsor the package for the entire months of October and November, but he'd like to tweak the package a bit. He'd like to only sponsor content pulled from FN (75% recipes) and FL and he'd like to add another ad unit to the mix.

7. Barbara returns to her saved package and adjusts the parameters of the brand pulls to match Bob's request. Once finished, she enters the campaign interface and tweaks the setting to display a big box ad unit. She sends the new preview back to Bob for final approval. Normally, Barbara relies solely on dynamic advertising to generate revenue, but holiday sponsorships are big sellers.

8. Bob gets back to her that same day and readily approves the campaign. With the contract process processing, Barbara now focuses on the elements that make up the final campaign. The placeholder ad units used in Bob's preview were drawn from a library that holds assets from prior campaigns. According to Bob, the only unit that can be reused in the new campaign is the logo. Barbara enters a creative request into the campaign generation interface, detailing the client's ideas for a seasonal campaign and the particular units that need to be created.

9. Once the client has approved the new creative, Barbara schedules the actual campaign. She returns to the campaign generation interface, chooses Fresh Market as the sponsor and assigns the proper mix of ad units to the campaign. Barbara matches the new creative assets to the campaign, and the system is ready to assign a sponsorship tag to the package level.

10. Barbara sets the campaign run-time and schedules the launch. With a single click, the sponsorship tag is generated behind the scenes, Dart picks up the ad unit assets and the sponsorship tag is applied to the package. All pages directly navigated to from the package page now present a package module with related links and the campaign ad units. Time to go find another sponsor.



BARBARA JONES
Create Package for Sponsor, Context Scenario

1. Barbara receives a call from Sandy Keyes at John Deere. She's looking to sponsor a summer package that emphasizes landscaping and she'd like to relegate the content to articles and video pulled only from HGTV and DIY.

2. After logging into the CMS, Barbara runs a query for "landscaping" in her customized metadata interface. The results show all content types tagged "landscaping," including data referring to the amount of specific brand and taxonomy results. Included in the return are co-occurrence tag results – associative, related tags to "landscaping," which provide opportunities to expand or shift the set of results entirely.

3. Barbara previews a straight "landscaping" package from HGTV and DIY. The article mix doesn't quite fill the template, so she clicks on a few related tags (e.g "yard" and "lawn") hoping that they'll provide better results. The package page refreshes and the results fill the page with even better relevancy than before.

4. Barbara generates a campaign unit preview for the package and the system provides Sandy access to review. Sandy is thrilled with the package and in return provides Barbara with the creative for the ad units. Once the assets are entered into the system, Barbara focuses on the final sell.

5. Barbara informs Sandy that the internal design team can craft a slick header for the package that represents the concept of "summer landscaping." Sandy reviews a few past crafted packages and determines that the template presentation has enough branding and messaging for her tastes and doesn't feel that the added cost would be a worthwhile investment.



DAVID MIMS
SEO QA, Context Scenario

1. David and his team have created a number of SEO best practices for Editorial to follow when publishing content. When David logs into the CMS, he is immediately presented with three separate views of Editorial output: latest published articles and modules, newly created package pages and top-level pages ready to go live and tags without synonyms.

2. David first engages with the river of articles that have been recently published. Each article in the feed is represented not as an article per se; each published piece presents the main SEO components of the article. Without serving as a bottleneck to productivity, David begins to scan for missed SEO opportunities.

3. With 45 new articles and 12 edited modules going live since yesterday, David scans through the first twenty with no concerns. Just as he hits the halfway point in his review, David notices an article in his feed that has two missing alt tags and a poorly crafted title. With permission to edit articles, David clicks directly through to the article creation page and makes the changes himself. He finishes scanning the published articles feed without needing to make additional changes.

4. In another view are two new top-level page redesigns waiting to go live. David is the last approval point in this particular workflow, so he follows similar steps as with the articles – reviewing the home pages to ensure that the project team SEO applied best practices when crafting the new templates. After reviewing the SEO elements in detail, David approves the pages and the system launches the new templates.

5. The last step in David's morning process is to review the latest tags being entered into the system *without* search optimized synonyms. Since yesterday morning, 36 new tags have been created across the

network properties. David has a separate program that can generate a short list of optimal search terms from a single or compound term entered into its system. David runs each new tag through this process and assigns the optimal terms as synonyms to the tags.



SARAH LUAN
Appropriation & Sketching, Context Scenario

1. Jeff Polinsky on line one. It seems as though Jeff has an idea to increase traffic on the site by changing up the home page. He said that he found a module on a network site that would work great in a new home page, exposing peer usage patterns as a navigational mechanism.

2. Sarah logs into the CMS and taps the module library. She looks up the module that Jeff referred to and after playing with it a bit, it does seem to make sense for the homepage.

3. Sarah imports the sister platform's module into her platform, creating a custom instance in their sandbox. She tweaks the layout, makes a few visual changes that aren't automatically available, and shares her specifications with Scott.

4. A ping is sent to Scott and Jeff. Jeff approves the design, and Scott quickly tweaks the module attributes, making them available in the design system.



SARAH LUAN
New Module Idea, Context Scenario

1. Jeff came to Sarah and Scott to discuss an entirely new type of module. After a few conversations, Sarah has enough information to start designing.

2. She details a wireframe of the proposed behaviors and sends it to Jeff for review. A couple of iterations later, the specifications are tight enough for Scott to produce a skeleton prototype.

3. In the meantime, Sarah begins to document the style elements for the module, aware of the brand style and pattern guide that she has implemented across other modules. She documents the explicit form elements that need to be represented in the style admin screen of the module and specifies the interface necessary for build.

4. Sarah logs into the CMS and not seeing a module project created by Jeff, decides to create one herself. She uploads her specifications to the system and adds Jeff and Scott as team members. After saving the project, it shows up on her dashboard queue as a project “In-Flight.”

5. Three weeks later, Scott completes the module and the corresponding admin screens. Jeff and Sarah review it on the staging server from within the module library of the CMS, and are pleased with the presentation. Scott is given the greenlight to make it available in the library.



SARAH LUAN

Tweaking Module Style to fit Brand, Context Scenario

1. Sarah receives an email alert that there is a module ready to be styled to fit her brand. The email is sent as an auto-notification when Jeff Polinsky saves a new instance of a module leveraged from another brand platform

2. Sarah logs into the CMS and finds a link to the module in question at the top of her work queue. She clicks on the link and lands in the edit area for the module.

3. Sarah first reviews the auto-style applied by the system. All style elements created by the original designer of the module are automatically cross-referenced against the CSS for the new brand. Where there are matching definitions (link color, headers, etc.), the system automatically applies the new brand style.

4. As it turns out, the system did a pretty decent job with this module, but there are spacing and border color elements that need redefinition. Sarah enters the style admin interface and tweaks the rogue elements.

5. She saves the changes, previews the presentation and designates it as finished, which automatically sends a notification to Jeff that the module is ready to use.



JENNIFER STEWART
Validate Submitted Tags, Context Scenario

1. Jennifer works outside of the CMS most of her day, collaborating with brand teams and SEO to ensure the domain taxonomy is as robust and exhaustive as possible.

2. Every now and then, Jennifer receives an email alert from the CMS, notifying her that an editor has added new, unique tags to articles category facets. Today, Jennifer has a few such alerts. It looks like whomever originally created the category facets for a new DIY show might not have specified enough choices.

3. Jennifer logs into the CMS and scans her dashboard queue for the newly submitted terms. She sees each term in context of the article type and title. One term looks like an obvious miss—the every other Friday co-host of the show wasn't included as a choice in the "Host" dropdown.

4. Before approving the additional tag, Jennifer double-checks the "Host" facet inline. As it turns out, the editor submitted the term "Kathy Jones" when "Cathy Jones" is the correct spelling and is already available in the facet. Jennifer adds the misspelling as a synonym to the correct term. She then removes the freshly submitted incorrect tag from the system and includes a note describing the circumstances of the removal. The system delivers the event driven note to the editor who submitted the rogue term into the system.

5. The other tag submission is the term "mahogany." Apparently, the very same DIY show covered a walkthrough for building a dining room table. The "Material" facet didn't have mahogany as a material choice for wood. Jennifer clicks on the title of the article to read the live article. Sure enough, the primary material used in the show is mahogany.

Jennifer approves the new tag with a single click and the system sends her previously created “approved message” off to the editor.

6. Something about that last approval struck Jennifer as strange. While the term definitely fit the facet, it seems as though controlling such a broad facet, as “material,” might not be a smart idea. If it were “wood,” possibly, but the facet seems too broad—this type of add is bound to occur again.

7. Jennifer begins a conversation with the DIY Editorial team and after a bit of back and forth, the teams agree that “Material” should become an uncontrolled facet—at least until Jennifer can suggest a number of precise substitutes.

8. Jennifer heads back into the system to the facet management area and searches for “Material.” She toggles its “controlled” setting to “uncontrolled” and calls it a day.



JENNIFER STEWART
Expand Facets, Context Scenario

1. Jennifer receives email alerts when an editor in the midst of crafting an article submits a request for a new structured facet to be enabled for a particular article.

2. It's been a few days since the last request has hit her inbox, but today one such request has made its way to her. Jennifer opens the CMS and scans her dashboard for the new submission.

3. In order to understand the context of the submission, Jennifer reviews the structure of the article entry interface. She clicks on the article type link next to the requested term, which takes her to the particular article interface.

4. After reviewing a number of past articles generated from this particular show interface, Jennifer can't seem to understand why the new facet is absolutely necessary. Time to take this conversation to the editor who submitted the request.

5. Jennifer emails the editor. After receiving an unconvincing explanation, Jennifer picks up the phone and gives the editor a call.

6. After the conversation, Jennifer better understands the impetus of the submission. It seems as though there are a good number of specific facets in the interface and the editor felt that if this particular degree of specificity has shown up previously, it would improve the collection of metadata and provide a useful degree of relational data.

7. In the end, Jennifer doesn't agree. Facets have to be carefully chosen on a continuum of broad to narrow classifications, but more importantly, Jennifer needs to keep the number of classes available on any given

interface to a manageable number. Editors (and Advertising) love to add layers of specificity, but hate spending more time publishing individual articles.

8. Jennifer returns to her dashboard and denies the request, adding her justification to a text field. The editor is alerted of the decision and the request is stored for posterity.

Key Path Scenarios: Mapping Form and Function

Key path scenarios are narratives that describe the persona using the system to accomplish their goals in the most probable and detailed fashion. From within these scenarios, interaction design solutions are formulated which lead to the design of the interface.

Key path scenarios provide the core framework for behavior and navigation of system. Derivatives and fringe case scenarios will be identified and carefully considered as needed.



ANGELA DAVIES, Editor

KEY PATH: Publish New Article

Input Copy

- Angela's dashboard displays saved articles, draft articles, recent articles published, etc.
- She chooses NEW ARTICLE from a list of article types, which is based her editorial responsibilities (specific brand, section, etc.) – ex., Interview with Emeril for Food.com
- Article displays document details - status of article, author, date created, version #, date, and when modified, date and author of last edit
- Angela enters copy in BODY field
    - Formats copy with rich-text editor (bold, italic, bullets, link, image justification, alignment, etc.)
    - Copy and paste from Word retains rich-text format and controls (curly quotes, paragraph breaks, bold, italic, link, paragraph alignment)
    - Typographical errors display as red links - words that are unknown to system; potential spelling errors (follow web form best practice functionality)

Add Tags

- Angela needs to categorize and map context to her new article using facets and tagging. She'll use:
    - Controlled facets – topics we know (baking recipe)
    - Non-controlled facets – anything new (Low-Fat Blueberry Pie)
    - Free-form tags – anything relative (healthy treats)
- Angela is presented with a number of CONTENT TOPICS (tag classes; controlled facets) and clicks the check box of each topic that is relevant to her new article
- She can suggest a new topic by entering it in the NEW TOPIC field and clicking SUGGEST TOPIC. System notifies Jennifer (Library Science) who begins vetting process offline
- Angela enters tags in non-controlled facets and system notifies Jennifer of the addition so she can approve or delete
- Angela enters free-form keywords in ADDITIONAL TAGS field
    - Spell checking is available
- Unwanted free-form tags are manually DELETED
- She clicks FIND PHOTOS and photo images that match article tags are displayed
- Image tags are displayed and Angela may add to or remove image tags to dynamically

alter the thumbnail image search results

Select Photos

- Angela clicks image thumbnail to view larger size
- She clicks larger image to reduce to return to thumbnail size
- She hovers over thumbnail to view images details – tags, description, date created, usage history
- She clicks ADD to place image ID where cursor is positioned in BODY field
- She deletes image(s) from BODY by removing ID
- She justifies image by highlighting image ID and choosing option from justification widget in rich-text editor

Upload New Photos

- Angela has 5 photos on her PC that she wishes to include in a new article
- From her dashboard, she clicks NEW and selects PHOTO UPLOAD in order to get them into the system
- The screen displays and Angela selects 5 images to drag into PHOTOS field
    - She may click ADD to place more images in this batch
    - She may select 1 or many images and click DELETE to remove from batch
- All 5 photos display as thumbnails as each one is processed
- When a photo is selected Angela may edit the default filename and enter a new TITLE, along with a DESCRIPTION and TAGS, for each photo
- Angela can apply batch TAGS for metadata that lives across all uploaded images (shortcut)
- DATA SIZE and DIMENSION are displayed when image is selected
- Angela selects the appropriate level of USAGE RIGHTS (e.g., stock photo, royalty-free, etc.) and checks the names of individuals she wishes to alert via email of a new photos
- She clicks UPLOAD to start process and progress bar displays
- When 100% complete a confirmation message displays “5 photos are now available”, and the PHOTOS field is empty.
- Each photo has a “page” in the system and can be found through SEARCH and INDEX PAGES by TAG
- All editable fields on upload are available to edit on the image page itself
- Each photo page provides access to resized versions of the original image based on height or width standards (portrait or landscape) squared across all brand needs
- The primary image page has an area that presents the trackback (usage) history of all sizes of the particular image

Save/Publish

- Angela elects to SAVE work in progress
    - Auto-save is enabled by default (every 15 secs)
    - CANCEL terminates all work in progress (and return to previous state - e.g., dashboard)
    - SAVE AS DRAFT returns user to dashboard; article not published, in draft queue
- Angela elects to PUBLISH work
    - She clicks PREVIEW to view article output in final format

- She can assign a publishing date in the future if desirable (return user to dashboard and article displays with publish date displayed)
- She clicks PUBLISH in preview mode to confirm article is complete, and status changes from DRAFT to Version 1.0, returning user to dashboard
- System sends alert to manager, brand product manager. SEO is informed when article appears in his river of published terms.



TINA GORDON, Freelance Writer

KEY PATH: Publish New Article
Same as Angela, but PUBLISH is restricted to Editor

Base Article Types for Publishers:

- **Article** – flat content
- **Recipe** – ingredients and instructions
- **Slideshow** – series of photos with editorial commentary
- **Quiz** – series of questions and user-generated answers
- **Show** – general info and on-air schedule
- **Episode** – instance of show on specific date
- **Profile** – general info on host, cast member, or special guest
- **Transcript** – text version of episode



JEFF POLINSKY, Product Manager

KEY PATH: Publish Content to Module

Modify Module

- Jeff's dashboard displays links to homepage, section fronts, and sub-section templates
- He clicks link to view template and reviews layout with modules displayed
- He clicks EDIT on a module to change it's content
    - Jeff may click DUPLICATE to make a copy of the current module – i.e., to create a second Poll, or to create a second list of recipes related to overall context of page
    - Jeff then enters a NEW NAME for this module and clicks SAVE
- Jeff enters new copy into fields, and alters other aspects of module – i.e., number of items displayed in a list, or sequence of images in a slideshow
- Jeff elects to SCHEDULE the module to publish on a particular day by selecting the date, which defaults to the current date, month and year.
    - If not scheduled, the module will publish immediately, on SAVE
- Jeff clicks SAVE to store the new content, and to return the module to it's original state

Save/Publish

Same as Angela



SARAH LUAN, Designer

KEY PATH: Template/Module Redesign/Alteration

- From her dashboard, Sarah clicks NEW, selects PROJECT, gives the project a name and clicks GO
- In the PROJECT TYPE area of the interface, she skips over NEW and selects EXISTING ELEMENT
- Sarah browses through the TEMPLATE or MODULE LIBRARY until she finds the element that is up for redesign/alteration.
- Sarah selects the element and clicks GO
- On return to the PROJECT screen, there is a link present to the element in the library, as well as its PRESENTATION CODE
- She moves across the interface and clicks ADD MOCKUP in the Visual Specs module to locate the FILE on her Mac for UPLOAD
- Upon upload, the file appears as thumbnail; Sarah adds a TITLE, DESCRIPTION and detailed instructions in the PRODUCTION NOTES field before clicking SAVE
- Sarah then assigns a TARGET DATE by selecting day, month, year and time.
- Sarah adds Scott, Jeff and a backend developer to the PROJECT
- Sarah clicks SAVE and returns to her dashboard where a link to the new PROJECT appears in her queue



BARBARA JONES, Ad Sales

KEY PATH: Sales Production

Create Package (content)

- She clicks or enters a keyword in module on dashboard to view search results
- Package results include articles, photos, and videos related to keyword(s)
- Barbara can dynamically control the percentage of articles, links, photos and video, using the refinement controls at top of results page
- Barbara chooses specific ad vehicles (big box, banner, etc. -- with name of sponsor in each ad)
    - Barbara clicks ad unit to enter name of brand and/or campaign which displays in ad unit when previewed
    - She clicks SAVE and enters text in each unit as desired
- She clicks PREVIEW to display desired results in package layout with ad units and all assets
- Barbara clicks SAVE PACKAGE and enters package name – e.g., Thanksgiving Package
    - She can elect to save multiple variations of a package with different percentages of content – ex., 50% photos, 25% video, 15% articles, 10% links to create the “Thanksgiving Media Package”

Create Campaign (creative)

- Barbara clicks NEW CAMPAIGN link to create a new sponsorship campaign
- She chooses the ad units for the campaign out of a list of ad unit types – e.g., checking big box, logo and banner, but not baseboard and text ads
    - If sales target is a new prospect, Barbara selects generic creative as placeholders, or selects rough creative concepts to apply to ad units
    - If sales target is a known client, Barbara selects creative from a pool of previously generated creative assets for desired ad units – i.e., Lowe’s 2006 campaign. New ad units will display generic placeholder to highlight new value proposition.
- She then enters NAME of new campaign– Lowe’s Thanksgiving Campaign 2007 – and chooses the package she saved – Thanksgiving Package – and clicks SAVE
    - Barbara can save multiple variations of a campaign with different creative assets

- Once campaign is chosen Barbara enters a name for the final package, which now consists of search results and the desired creative campaign for all ad units, and she titles it “Lowe’s Thanksgiving Package 2007”
- She clicks SAVE and a confirmation screen presents a PREVIEW PACKAGE link
- Barbara clicks the link and the package layout displays in it’s final format (as it would appear if it were live) with appropriate campaign ad units in place
- She clicks SEND TO CLIENT and email field pops up requesting user to enter client’s email address(s) and a short personal message
- She click SEND to deliver message, which includes access info and link to mock package, requesting s/he securely review mockup online
- Barbara may EDIT package (search results) and campaign assets (creative) to client’s desire

Launch Sponsored Package

- Barbara clicks title of proposed campaign – Lowe’s Thanksgiving Campaign 2007 – listed as PENDING in module on dashboard
- She clicks BROWSE next to chosen ad units to locate approved image to replace previous mock creative (repeat as needed throughout package layout)
- She checks the preview link to see the new assets attributed to the correct package
- She clicks GENERATE SPONSORSHIP and processing bar appears
- When 100% complete, link reads as SPONSORSHIP TAG ASSIGNED
- Barbara clicks SCHEDULE to specify date/time of “Run Time”
- On SAVE, a message is sent to the client once again allowing them to review the package layout in its final form
- On receipt of client approval, Barbara returns to campaign interface and clicks LAUNCH PACKAGE button
- Package goes live (or pending based on schedule) and an alert is pushed to the appropriate internal staff’s dashboards



SCOTT McBRIDE, Content Engineer

KEY PATH: Create/Add New Template

- Scott finds new template PROJECT link in 'pending' dashboard queue
- The Project screen presents designer's specs and a list of all corresponding modules with links to their presentation code
    - Jeff and Sarah upload a screenshot as a "proof" of the final design; Scott's spec file to reference during development
- Scott develops the final code for the new template offline
- When complete, Scott returns to the CMS and clicks on the project link
- Once in the PROJECT screen – view of all things that make up the new template – Scott navigates to the FINAL CODE area and clicks on BROWSE to select the final code document from his PC
- He UPLOADS desired document to the project page
- Scott previews the code on the staging server
    - If the code needs adjusting, Scott has access to the raw code in a view within the CMS
    - If the code is good, Scott SAVES the project with the final code
- Scott clicks on SEND ALERT to notify the project team that the code is complete and ready for review. Jeff (PM), Sarah (Design) and David (SEO) must approve the preview before Scott can schedule the template to launch

KEY PATH: Create New Module

- Scott finds new module Project link in PENDING dashboard queue
- Once in the Project screen – with the design specs and details in plain sight – Scott navigates to the FINAL CODE area and clicks on BROWSE to select two code documents from his PC:
    - Presentation code for the actual module
    - Admin screen for content management and visual styling
- He UPLOADS desired documents to the project page
- Scott previews the output of each document on the staging server
    - If the code needs adjusting, Scott has access to the raw code in a view within the CMS
    - If the code is good to go, Scott saves the project with the final code

- Scott clicks on SEND ALERT to notify the project team that the code is complete and ready for review. Jeff (PM) and Sarah (Design) must approve the preview before Scott can schedule the module to appear in the CMS module library