Mapping the modern WEBDesign process

1. Web Design as a process

- Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design, interface design, authoring, including standardized code and proprietary software, user experience design, and search engine optimization.
- What is the goal or purpose of web design? Looking at a website, what do you see? Text, images, video, audio, and sometimes interactive elements, all sourced from a server and presented to you on a screen or through assistive technologies. And if you take a moment to examine the nature of all of these elements, you'll realize they're all representations of the same matter, information. So, a website is a means of presenting information. And what makes the web unique in the way it presents that information, is our ability to link different pieces of information together and change and customize the user experience of accessing that information. That means at its core the goal or purpose of web design is to create informational user experiences.
- Viewed from a distance, we can see the Modern Web Design Process as a series of discreet phases. Each of which focuses on a task or group of tasks. And each of which has a significant role to play in the creation of the final product.

2.Some Perspective

- For site owner technical implementation, platform, ease of publishing, ease of interaction is important.
- For site user findability, informational value, accuracy and shareability are important.
- Your job is to find best possible solution for both.
- End user is only interested in accessing the content on the site. It does not matter which operating system or which device for them.
- Web designer’s feelings on the end users’ choices are irrelevant. His duty is to provide the best informational user experience to end users, regardless of how they choose to access the information.

- Adopt progressive enhancement.
- Design around content.

3.Preprocess

- Typical roles in a web designing project are content creator, content strategist, information architect, user tester, graphic designer, user experience designer, front end developer, back end or CMS developer, system architect, quality assurance lead and beta testing supervisor.
- OfCourse one person can do all or at least multiple roles.
- Three truth about web design are:
- A website is a living thing. By nature, it will evolve.
- Building a website is usually a team effort.
- A website is rarely built in a vacuum.
- Create guides for content, style, code and process.
- Version control is a technical term for creating snapshots and virtual timelines for your content.
- Without version control, everyone works on parallel versions of the project.
- Everyone ends up with different results.
- In version control new components and experiments are branched out and merged back only when they are completed.
- GitHub is best for version control.

4.The Content Strategy Process

- Content strategy is the planning, development and management of content for the web.
- The main purpose of website is communication with end users and for that purpose content is very important.
- Create personas.
- Ask question like that “who do you want to reach with the website”?
- Personas should base off real end users.
- Create at least three personas for any project.
- We've established that content is all that really matters. So, it naturally follows that every web design project should start with the content. What we need to do here is create inventory, and audit all content, and list every

asset and how it relates to other assets.it is called, "the content audit." "asset "is referred to each item that may appear on the site. Either on its own, or in conjunction with other content. That may be a page, or it may be text, images, video, or other elements that are displayed together on a page. The idea here is to identify every asset and make sure it's accounted for as the project progresses. The content audit is crucial, whether you are starting from scratch, or have an existing website that is being rebuilt.

- Once the content audit is complete, you'll have a clearer picture of what types of content will be presented on the site. Looking at the page assets, you can now start identifying different types of pages. And from there, establish discrete content models. It's easier to see how this works with a practical example. So, let's say we're building a site for a food services business called Eat. The company provides various services including catering, cooking classes, a chef-at-home service and a restaurant. To promote all these services and build awareness of the company, the website will have several different types of content. Standard pages including landing page, an about page, contact page, legal information, and so on. A services section with an index and separate landing pages for each service. A recipe section, where the owner publishes free recipes and cooking tips. A blog, with customer stories and promotions, and general news items. And finally, a testimonial section linked to the different services. Based on this list we can tentatively identify three different types of content that fall outside the regular website content models. Services pages, recipes pages, and testimonial pages. Next, we need to build out the content models for each of these types to see how they differ. This process involves taking each item in turn and asking, "What information goes on this page?" For the services page we have the service title and service description. And we also have promotional items and related testimonials. For recipes we have the recipe title, we have ingredient lists, recipe metadata like cooking time, difficultly level, portion size, and so on. A general description, step-by-step instructions and maybe a gallery. And for testimonials we have the name of the person, title and company affiliation, maybe a headshot, the testimonial itself and also a way of relating the testimonial to one of the service pages. Having mapped this out, we now know we have five different content models that can be attributed to the assets in the content audit. Regular pages, regular blog posts, services pages, recipe pages, and testimonial pages. The creation of content models will inform our choice of content management system. And also help establish how the system will be set up.

- We know who the end users are and we know what content will be displayed on the site. The next step is to organize that content based on priority hierarchies. There are two levels of hierarchies. Site-wide hierarchies and individual-view hierarchies. Site-wide hierarchies have to do with the overall structure of the site. If the visitor lands on the site, what content should she see first? The front page is a likely top tier candidate. Index pages are other likely candidates for the top tier spots. Individual blog posts and recipes will likely live on the second tier. They may be linked to and landed on individually, but their primary purpose is to drive the visitor up to the top tier and then onward to the services where the goals of the site owner are being met. Legal pages and other static information about the company and its employees live on the third tier. These pages are important, yes, but only for a visitor who is already interested and wants to learn more about the company. These pages and their content are unlikely to provide direct value for the visitor or site owner and are unlikely to play a deciding role when a purchase or acquisition is considered.
- The process of creating content priority hierarchies is naturally followed by the creation of information architecture for the site as a whole. The term information architecture, or IA for short, is a bit of a weird one because it refers to three different, but connected thinks. The actual architecture of information. The act of creating that architecture. And, the community of practice that thinks and innovates in this realm. he idea of information architecture is really quite simple. We want to organize and label the different components of a website in such a way that a visitor or user can get from one place to another and find what she's looking for in the most obvious and straight forward way possible. Easy as pie. So, how do we do that? Well, taking all the information we've collected so far, we can now draw up a diagram that maps out each of the main elements of the site. That would be the different pages or views and the different content models and shows how they are displayed in hierarchy and how they connect to one another.

- Now, it's time to start pulling all those pieces of information into layouts to see how they would fit together on real screens in the real world. What we are aiming for here are crude wireframes, essentially boxes with content that are devoid of design and visual elements. Think of the wireframe as a blueprint for a website. It outlines the components and shows the content in a real-life context without distracting elements like colors, fonts, and decoration. And like the blueprint, the wireframe can have annotations. Arrows, lines, circles, and comments that explain and provide further information about what's going on. The purpose of the wireframe is to lay out the overall structure of each of the pages and views on the site. By building a wireframe for a page, you can visualize how the different components will interact and where they will live relative to one another. The wireframing process should start with the narrowest mobile screen, currently a vertical iPhone 5 and be structured based on the content priority hierarchies established earlier.

5.Prebuild Testing

- User testing is an essential part of the web design process. Your client and anyone associated with the client will be heavily biased by existing knowledge of the subject, company, and goals of the website. They've also likely been exposed to anything the company has produced and, in all likelihood, also the web design process itself.
- the purpose of the Card Sorting exercise is to get the end users to organize the website content into hierarchies and information architecture they feel is intuitive. The resulting data can then be used to revisit everything from navigation labeling to content hierarchies, to the proposed information architecture of the site. Here's how the Card Sorting exercise works. First, decide on a scope for the exercise. Are testing the content hierarchies of pages? Are you testing category and taxonomy structures? Or are you testing the information architecture for the whole site? Next, take a stack of cue cards, and write the title of one piece of content on each card. If you are testing for content hierarchies on pages, you'd put the different pieces of content on each card. For taxonomy structures, you'd put the items that fall within the taxonomies on each card. For the information architecture of the whole site, each card will represent one page on the site. Place the

cards in random order on a large table in front of an end user. Instruct her to sort the cards in whatever way she thinks is logical. Provide her with blank cards and a pen to create additional cards if she feels something is missing. The user is free to do whatever she wants with the cards. She can organize them, discard cards she feels are irrelevant, and create new cards or structure where necessary. Observe the user, and have her explain her thinking as she progresses through the exercise. When the sorting is completed, have the user name each section, if she creates new sections, on a new card, and take pictures for yourself of the entire layout. From here, you have several options. If there is time and number of cards is under 50, you can do a re-sort, where you scramble the cards and have the end user do the exercise again. This will usually produce a new and refined ordering. If you have multiple end users doing the exercise at the same time, you can have them switch tables and re-sort the work of someone else. Throughout the exercise, make sure you take copious notes of what the user says, and take pictures of the card ordering for later analysis. Once you've performed the Card Sorting exercise with several different end users and representatives from the clients, you'll usually see patterns emerge, and you can use this data to refine your content hierarchies, your information architecture, and your general understanding of the project as a whole.

- In addition to the card sorting exercises, you should also test the preliminary wireframes and plan interaction patterns on end users. Present the end user with paper versions of the wireframes and have them access the content and pretend to navigate through the different sections. If they navigate to a new page, swap the pages. If they activate an interaction pattern like a hidden menu on a mobile device, have a paper version of the menu overlap the section it would overlap on the website. Essentially, you are mimicking the real-world behavior of the final product as you currently envision it. Here it's vitally important that the content on the pages is real content from the site. You want the end user to interact with the content as if she was accessing it in real life and this can only be done by presenting her with the actual content itself

6.The build Phase

- Design-->development-->content creation
- Our build phase can loosely be split into three discreet tracks, content creation and curation, graphic and user experience design, and front and back-end development. The content strategy established during the content strategy and pre-build testing phases, provides each of the tracks with a solid foundation from which work can now be started. By running the processes in parallel with a staggered timeline, underlying structures for content can be built, content can be added, and designs can be applied in an ordered fashion that keeps everyone focused on the same aspects of the site at the same time.
- Style guide describes how all elements should be styled, to provide consistent visual communication.
- At its core, the style guide is a document that describes through words, numbers, and examples, how all items are to be styled and designed to produce consistent visual communication. When a style guide is complete, any member of the team, old or new, should be able to build new components or change existing ones based on the rules outlined in the document. Style guides have been around in the print world and other media for decades, and are now becoming a more common component of the web design process as well. A style guide for the web differs from one that targets print or other media in one crucial way.
- Style guide is a living document, should be developed in tandem with the project and should evolve with the site.
- When the guide is complete it should act as go-to reference.
- Guide should be in version control system to track recent changes.
- To get the build process started, and all the tracks running, the development team needs to build the foundation for the content models. This means creating whatever structures are necessary in the site or the CMS so the content team and client can start inputting content and see that content displayed on the front end. If you're working with a CMS, this process will be made easier by creating a template map that corresponds with the information architecture of the site.
- By identifying what templates are going to power what pages and which content models applies to those pages, the development team has a clear

plan of action to get the core content models developed. Using WordPress and our imaginary business EET as an example, this would mean building out a template map that identifies the different custom polls types necessary and then creating the actual custom polls types as plug-ins or within the custom theme. Once the customs polls types are created, custom taxonomies and fields are added and front-facing templates are built to display the default content fields as well as the new custom taxonomies and fields.

- As the development team begins its work, it's important to affirm the Baseline Development Priorities of all websites. These priorities should be universal gospel and common knowledge for all web designers. But even so, they are worth repeating every time a new project begins. Think of the Baseline Development Priorities as a mantra you say to yourself as you work. Maybe even print them out and paste them above your workstation. They go as follows, one, all code must follow web standards. Two, all content must meet accessibility guidelines. And three, all design and development must be "mobile first". These three rules ensure, a, that your code is written properly, is maintainable, and will work across all browsers, b, anyone can access the content in any way they want or need, and c, the most likely method for accessing the content is the method everything is designed and developed for.
- To ensure the content hierarchy is adhered to and that accessibility is considered from step one, the first step in the development process should be to create a content-only build, a website that displays the content in a logical structure and order but without any style. In technical terms, this means building a website to display all the content in raw HTML first. There are two key reasons for starting with a content-only build. The first and most obvious one is that you start off with a solid and properly built foundation. By focusing on HTML structure and content first, you ensure that the code of the site meets web standards and that even if all styles and JavaScript fail, the content is still available and navigable for the end user.
- responsive web design is a process much like web design as a whole and that process has some fundamental constants. The first constant is mobile first.
- Design the mobile first.

- Add media queries that belong near the original element.
- There are no predefined breakpoints.
- Following the three constants of responsive web design, the actual job of designing your site across all possible screen widths is best done in the browser. You design for the medium the final product will be presented in, and that medium is the browser. In practical terms this means we build the design from the ground up just like we did with content and accessibility: one piece at a time, in code. To web designers who are used to doing the visual design in Photoshop and then transposing it over to an IDE, this will be a bit of a transition. That said, there are major benefits to the in-browser design process that will make you less dependent on Photoshop in a very short time. And of course, you can still use other design tools to help this process along, especially to make mock-ups for clients. But the majority of your design work should be done in the browser.
- For browser design, you should have a content-only html site.
- Add CSS to bring it in line with your style guide.
- Focus on one content block at a time.
- Add block information to your style guide for future reference.
- Progressive enhancement is all browsers display a full featured baseline experience, while advance browsers display extra features to improve the experience further.
- To make images and graphics fully responsive we have the new source set and sizes attributes for the image element as well as the picture element. Combined, these allow us to serve up different image files depending on screen size, resolution, and bandwidth to meet the demands of the modern browser market.
- For icons, we can now use custom icon fonts. These graphics allow us to apply font-specific CSS to flat icons and serve up scalable icons that look crisp and clear on all screens and all resolutions, and because they are fonts, they are light to load and easy to render for the web browser. For more complex graphics and animations, we can now use scalable vector graphics or SVGs. The amazing thing about these graphics is that they are individual XML documents that draw lines and curves based on numbers and mathematical formulas. That means we can style their individual

elements using CSS and, more impressively, interact with them and change them using JavaScript.

7.The test and Revision Process

- Test and revision phase includes:
- First Encounter
- Accessibility
- Break Testing
- Redesign
- Rebuild
- Test
- Feedback
- First round of testing should be accessibility testing.
- Accessibility testing is part technical, meaning you and your team can do some of it in your own office without having to bring in end-users. Here I'm talking about baseline testing for web standards and accessibility compliance, keyword navigation, image and media tags and content hierarchies. To test the code on your site run it through an accessibility validator like achecker.ca. This tool will provide a detailed breakdown of how your markup and contents stack up to accessibility guidelines. It is also a good idea to run it through an HMTL validation check at the same time to clean up any errors lurking in your code. Next, visit the front page of your site and turn off your mouse. You should be able to navigate to any section of the site using only your keyboard without difficulty. Here it's important to pay attention to focus states of all clickable elements including links and linked images to ensure the visitor can clearly see what element is currently in focus and available for a click action. To test content hierarchy, disable all styles and JavaScript and read the page from top to bottom. There should be a natural hierarchy in the content when it's displayed in its raw format. When the technical tests are done put the site in front of real users with accessibility issues. Ideally, you want to test it by people with visual impairments, motor problems and cognitive disabilities at a minimum.
- When accessibility testing is done and any accessibility issues have been addressed and remedied the next step is what I call Break and Edge-case

testing. Here you are submitting the site to scenarios where things don't work the way they're supposed to or the user does something unexpected. This can usually be done in-house but also benefits from some third-party testing. Break Testing, consist of literally breaking a component of the site and seeing how it behaves. This may mean disabling one or more style sheets or JavaScript, or it may mean loading the site on a bad connection, or a device with a broken browser. What you are testing for is how well the site performs under suboptimal conditions and whether the informational aspect of the site carries through even when parts of the site fail. Even without style sheets and JavaScript the content should still be accessible and consumable.

- With all technical tests out of the way, the site should go through a series of end-user tests. Just like the pre-build tests, this involves sitting an end user down with the website and having her consume the information and navigate through the site on her own. Here it can be useful to use screen capture software and a video camera to record user behavior on the site as well as user reactions and commentary in real life. If video is recorded, the user needs to sign a release form for you to be allowed to use her likeness. The end-user test should be done on various platforms, browsers, and devices to ensure the end user understands and is able to use the site regardless of how she chooses to access the site. It may be a good idea to start some users on mobile devices while others start on full browsers. It can also be useful to have the end user use her own devices or computer to access the site to test for unusual browser configurations.

8.The Optimization Phase

- Optimization priorities are:
- User experience
- Shareability
- Findability
- The primary goal of the optimization phase is to optimize the user experience of the site.
- Leadtime, responsiveness, accessibility, findability is very imp.
- Minify unused CSS.

- Images should be optimized.
- Svgs should be used.
- Cashing should be used to reduce server-side load.
- CDN services should be used.
- Optimization for social sharing involves design, user experience, developments and content creation and is an ongoing challenge. The social medial landscape changes constantly and so should your social media strategy. This is why leaving the social media sharing to a third-party service or plugins or extensions may be a wise choice than going it alone in the long run.
- the search engines index your site based on some familiar attributes: Quality of content and "trust factor", Accessibility, Load time, and Quality of code.
- Add alternative text to images, where appropriate.
- Add attachment pages and text to images.
- Submit dynamic sitemaps to search engines.
- Activate webmaster tools and site verifications.
- Write great content.
- To monitor the effectiveness of your content, design, development and marketing, analytics should be deeply integrated into your site. Analytics will give you hard data on visitor behavior on the site and can help you uncover positive and negative patterns that can be addressed. What type of analytics will work best for your site is an open question. There are many analytic providers to choose from and they all provide slightly different services. Adding multiple analytic services and cross referencing the results can be quite helpful in getting a clear picture of user behavior.

9.Launch and Reset

- All inventoried items encountered for
- All content signed off by owner.
- All content proofread
- All pages visited and read
- All menu item tested
- All links tested

- Interactivity tested.
- Site works on all platforms.
- Color contrast meets accessibility
- Font weight and sizes meet accessibility guidelines
- Site s key board accessible
- 404, nothing found and empty search encountered for.
- Valid code
- Cross browser compatible
- The exact procedure for the launch of a website depends on many factors. Is this a new site on a new domain or a new site replacing an old one? Is the launch of the website considering an event or is it a build it and they will come kind of affair? The list goes on, and the answers to these questions are project-specific, so I can't give you a one size fits all process here. Instead, we'll look at a launch process that can and should be applied to all sites, regardless of scope, size, or location. It consists of two stages, the soft launch and the hard launch. What exactly these two stages contain will vary, but the basic idea is always the same. The soft launch is a limited access launch for a select group of end users. In the software development world, this would be considered an RC, or release candidate launch. You're happy with the site and confident it will meet expectations, and now you want to test it on the real users before releasing it into the wild. The key to the soft launch is to limit the number of users able to access the site and collecting and analyzing the data they produce. This could be done in the form of analytics or direct questions through surveys. The soft launch is also the time you give access to the site to the client and all of their employees. Invariably, the client will have second thoughts about the site and will need time to adjust to the new reality of their site. There are also always issues that only appear once the site is live, and they start approaching it like a normal website, rather than a project they've been working on. By adding the client to the soft launch, you can address these issues before the site goes live to the world and also allow the client and their staff to familiarize themselves with the site in advance of the regular end user. The group of end users, who get early access through the soft launch, could be existing customers, eager fans, family and friends, even media. Announcing the soft

launch as an invitation-only prerelease will usually gather a lot of interest and result in a list of users who will provide good feedback. The tech blog, The Verge, did just such a prerelease of their new responsive design in September of 2014. They initially asked for 50 testers and ended up with several hundred. By making a big deal about the prerelease, and keeping the number of users small, they got focused feedback that could be used to finalize the product before the hard launch. This also showed the final benefit of doing a soft release. When a limited number of users have early access, they will invariably talk about it in social media to their friends. This should be encouraged, as it is a great way of building hype about the upcoming hard release. Just remember, you're looking for real end users here, not community influencers or trendsetters. Pre- and soft releases have gone very wrong when groups of social media users decide being part of the soft release is a status symbol in and of itself. That's not what you want. Unless something serious is discovered during the soft release, it should be followed by a hard release within a couple of days or weeks. The hard release is pretty self-explanatory. This is when you push the big red button and take the site live to the masses. To make this process as seamless as possible, I usually do two things. First, I build the site in a publicly hidden folder on the live server, well in advance, to ensure the launch is not hindered by DNS issues. Second, I take the site live early to select collaborators, to ensure everything is working properly across multiple networks. To do this, I use the combination of IP mapping and redirects. Set an official time for the release, and make sure you hit it, by starting the process in advance. It is amazing how angry clients and end users get if the site does not launch exactly when promised, and without starting the process in advance, even minute changes can cause unwarranted delays. During the hard launch, make sure all hands are on deck and ready to tackle issues. This includes testing, to make sure the site is actually live, fielding panicked calls from the client and confused end users, and creating a feedback loop for tackling issues. We'll talk more about that last point in the next movie. The main takeaway for the soft and hard launch is the same as always. Make a plan, complete with contingencies, stick to it, and everything should run smoothly.

- When a new site is launched it's important to have a feedback loop for issues and resolutions in place. There will always be issues that need to be addressed and many of these issues will be discovered by random end users visiting the site. During the first several months of the new site's life the end user must be provided with an easy way to report issues, including sending screenshots so these can be triaged and dealt with appropriately. In many cases the reported issues will not be actual issues but rather unfamiliarity with the new site but they should still be cataloged and addressed. It's essential to make the feedback process a Feedback Loop. A visitor reports an issue, the issue is triaged and addressed, and the visitor is provided with an answer. By keeping the visitor in the discussion, she will feel like her voice is being heard and she's not talking into a vacuum. And even if the issue reported is not fixed, she will feel positive about her engagement with the company.
- Once the site is launched and working properly, you need to run another round of user testing to ensure that everything is still working as expected. Here you should run tests on three key groups of users. Users that have never visited the site, new users with experience with the new site, and older users with experience with both the old and the new site. As before, you're testing for user comprehension, communication and accessibility. The three groups cover three different types of users who will give different types of data. The first group will tell you if the site is intuitive to use and meets it's communicative goals. The second group will tell you if there are latent issues and sticking points in the site that only become apparent once you've been using the site for a long time. The third group will tell you if users are able to transition from the old to the new site, and put-up flags if there is content or functionality that used to be available but is now either not present, or hard for them to find. This round of user testing can be informative, and may uncover issues that weren't redesigns, redevelopments, or even rethinking of whole sections of the site. For this reason, the post-launch user testing should be seen as part of the broader scope of the web design project, and the results should be used as planning starts for the next version or evolution of the site.
- Web design is a continuous process of constant iteration, innovation, and refinement. There really is no such thing as "one and done" in the web

design world, or at least there shouldn't be. The second your site goes live, you start the process over. Redefining content strategy, implementing new ideas, testing, iterating on existing content, designs, and code, optimizing for better performance and usability, and relaunching the site again and again.