

---

**process.ngenworks.com**  
**Documentation**

*Release 1.0*

**nGen Works**

December 01, 2014



<b>1</b>	<b>How nGen Works</b>	<b>3</b>
1.1	The Take Away . . . . .	3
<b>2</b>	<b>Table of Contents:</b>	<b>5</b>
2.1	Phase Zero: New Business . . . . .	5
2.2	Phase One: Discovery . . . . .	8
2.3	Phase Two: Definition . . . . .	11
2.4	Phase Three: Design . . . . .	14
2.5	Phase Four: Development . . . . .	22
2.6	Phase Five: Quality Assurance . . . . .	25
2.7	Phase Six: Delivery & Launch . . . . .	26
2.8	Ongoing: Maintenance . . . . .	28
2.9	Glossary . . . . .	29
2.10	Roles & Responsibilities . . . . .	30
2.11	Other Helpful Information . . . . .	34
2.12	Templates . . . . .	42



Contents:

- How nGen Works
  - The Take Away
- Table of Contents:



---

## How nGen Works

---

A process guide. Yep. We made one—and it's a good thing we did. Here you'll find a high-level view of all the phases, steps, and roles that make up the nGen Works approach. When you come to the end, stop. You should then be all knowledgeable.

We're built on flexibility and collaboration. Talk to your team and follow the process that works for all of you.

### 1.1 The Take Away

This is an ever-changing guide that should highlight how your expertise fits into the day-to-day. If you're ever unsure about how or where you fit in, just connect with your team and find out. We have each other's back.





---

## Table of Contents:

---

### 2.1 Phase Zero: New Business

Contents:

- Making Contact
  - Screening Call
- Call For Teams
- Chemistry Call
- Follow-up
- Getting The Scope
  - Estimating
  - Timelines
  - Research
  - Calculate
  - Chunk the Data
  - Prepare to Deliver
- The Paperwork

#### 2.1.1 Making Contact

A lead comes in through phone, email, or referral.

##### Screening Call

New biz team sets up an investigatory call with the prospect to see if we'd make a good fit. New biz can determine if the project seems viable, if the client will be fun to work with and may have funds set aside for the project. The team acts as gatekeepers to vet leads as they come in.

**YES** New biz team high fives, debriefs, and sets up call for teams.

**NOT US** If things look good but for some reason we can't take the lead, we send a polite "would love to work with you but the timing isn't right" email. If there's a shop we think could help the prospect we provide contact information for the prospect to initiate contact.

**NO THANKS** If things don't look good (red flags, ridiculous budget, or nonexistent timeline), a "thanks but no thanks" email is sent to the lead.

## 2.1.2 Call For Teams

If the new biz team feels good about the lead, they **send out an email** to the team about it: who the prospect is, what they need, and any other important info gleaned during the screening call. When working on a large lead, a start-up, or a completely undefined lead, sometimes it's impossible to determine who should be on the team right off the bat since roles have yet to be defined. The lead is sent out to both the core team and our Friendgeneers with the understanding that the core team has first dibs on new projects coming in the shop. It's a good idea to welcome teammates fitting anticipated roles that are likely needed, and then re-evaluate these roles together throughout the project. It's also a good idea to assign project managers for projects early on, so one person can keep a high-level view of the project and do the communicating on the team's behalf. The goal is to vet the lead to make sure it is a good fit for nGen.

nGeneers have **two days** to respond to the email. Responding means they dig the lead or want more info before committing. Responding with a thumbs up also means they'd probably like to work on the project, although they don't have to. If no interest from specific needed roles is expressed, others can stand in for a call and relay info back while the team looks for someone to fill the role. If there are not enough nGeneers to fill all the roles the project might require, we invite Friendgeneers to work with us on the project.

If, after two days, nGeneers express that the lead doesn't feel good, we send a "thanks, but no thanks" email to the prospect.

## 2.1.3 Chemistry Call

A conference call with the prospect is scheduled. The project manager leads the call and helps guide questions to determine the needs, goals, and complexity of the project and notes how the client interacts with the team. Team members can introduce themselves and assess the prospective client and project, asking any additional questions. It's important for the team to go into the call prepared; with their relevant questions based on the known project information in the project manager's hands so he or she can lead the call.

---

**Note:** It's a good idea to assign one note taker for consistency and, whenever possible, record the call.

---

## 2.1.4 Follow-up

A debrief follows right after the call. If all things shout "go", the new biz team works with the project manager and the team to determine a team champion—the team member who will take the lead role for design and/or development—to set up next steps.

If the lead reeks of red flags, new biz sends a "thanks but no thanks" email.

## 2.1.5 Getting The Scope

Getting a solid handle on the scope of a project involves a couple of steps. The project manager works with the rest of the team to analyze the scope of the project. Any roles that are in need of a team member can be analyzed and filled from a list of Friendgeneers.

### Estimating

We have started estimating time in weeks, not hours. This way, we can determine our team availability according to the percentage of time not already allocated to other projects. So for example, if we know a team member needs to devote 30% of the week to one project and 30% to another, we say they are 30% available that week for additional work. Our teammate can estimate how many weeks their portion of work might take if they only have 30% of each week available to a particular set of tasks. From there, we get every project teammate's availability and their estimate

on the number of weeks they need to finish the project including time for approvals and turnarounds. Estimating in weeks instead of hours gives us a fairly accurate idea of the total number of weeks a project should take and gives us the flexibility to adjust this timeframe if client approvals or deliverables come in late. It also prevents us from over scheduling team members and helps us maintain a high-level view of our overall availability.

We're working hard to reduce the number of projects our teammates are working on at any given time so that they can stay productive without being spread thin across too many tasks. Our ultimate goal is to reduce the number of projects each team member has on their plate to no more than three (two billable, and one internal). We're also very excited about our team-creation project called Jellyfish Jobs which should be ready soon. It will let us form virtual teams with core members and Friendgeneers when new projects come in and let us track how many active projects each teammate has at any given time.

## Timelines

Timelines on a project always change. Sometimes, through no fault of their own, a client will need more time to get approvals or other deliverables ready for our team. An important thing to note is that with weekly estimating, when our clients take longer than anticipated to complete tasks, we add additional weeks to the budget to accommodate the delays. Because this time can add up quickly and blow the budget, we protect our clients and ourselves with a "Pause Clause" in our proposal which says that we can put the project on hold and reschedule it according to our availability if deliverables are two weeks late. That way, no work is done on the project (aka no budget is used up during the pause) and the client now has some time to complete approvals or upload assets and deliverables.

If we've underestimated the number of weeks it will take to finish the project, we re-estimate internally as soon as we're aware, and let the client know right away. Over-communicating is always better than under-communicating. If you're thinking of trying weeks instead of hours, try estimating both ways and compare. If the numbers are wildly off, it might be time to adjust how you are billing or how accurately you are estimating time. Another way to strengthen your estimating is to have other team members weigh in on the estimate. That way newer teammates or people who are not as confident with estimating have supports in place to help them be more accurate. At the end of the day, we don't feel comfortable telling our teammates how much time they should take to do a task. They need to be the ones to share that with the team because we all trust we're doing our best work and we want to help keep each other accountable.

---

**Note:** A separate discovery can really help you out here. Although a final budget may not be clear until after separate discovery, determining a rough ballpark of your budget is important to both parties, especially since these numbers almost always evolve. It will help everyone know if it's feasible for the project to proceed.

---

## Research

If we aren't doing a discovery phase, we gather all the information we can to determine the estimated timeframe and budget. This includes things such as:

- overall project and business goals
- obstacles and bright spots
- stakeholder info (how many, what type of involvement, availability)
- content types
- current and desired technologies and functionality
- is this an app or site; which platform
- any anticipated external/third-party needs (DNS, hosting, email, integrations, newsletter vendors)
- if the project is required to work in outdated browsers
- extraneous concerns (marketing, branding, phases, business strategy)

- required components

## Calculate

More than one stakeholder? Increase the scope. Tight timeline? Increase that scope! Multiple integrations or an in-house staff? You got it. For every layer of complexity, it's best to increase the scope with extra time.

Andy Rutledge created a [good resource for defining scope](#), although it calculates hours not weeks. Consider using it as a guide to defining the size, projected timeline, and cost of your project.

## Chunk the Data

Once we have an idea of all the bits and pieces, we have every teammate estimate how many **weeks** should go into his or her portion. That way, rather than simply breaking the scope into phases, we have a solid idea of all the components and how they fit together. Once this has been done, we can allocate time to phases for things like asset collection, content writing and approvals, and holidays or vacation if needed.

## Prepare to Deliver

Make sure every team member has had a chance to weigh in on the estimated scope of the project before sending it over to the prospect to help us steer clear of miscalculations.

### 2.1.6 The Paperwork

The team works with the biz dev team to create a short proposal. The proposal should highlight who will be on the team, the expected scope, budget, and timeline. Ask nGeneers not on the team to review what you'll send and how it will be formatted.

Now that the proposal is ready, the team packages it with a Terms Of Service agreement and presents it to the prospect. These two documents seal the deal. Then you await approval. Have the prospect sign and date both, and return copies of each.

WAIT! SIGN-OFF REQUIRED!

---

**Note:** Numerous requests for revisions to your proposal can be a red flag that indicates the client wants to drive. Don't get sucked in to more than one or two revisions unless they're warranted!

---

Once both documents are signed and received it's time to initiate a deposit invoice and the prospect officially becomes a client. Lori asks the client for contact info for each person on their side then sets up the project in Basecamp and Harvest, sends a deposit invoice, imports any past communication—including the signed documents—and we're off! (Well, once the deposit payment is received, that is.)

## 2.2 Phase One: Discovery

Contents:

- Separate Discovery
  - The Usual First Steps
  - How it Works
    - \* Audit
    - \* Analysis
    - \* Strategy
- Traditional Discovery

## 2.2.1 Separate Discovery

A separate discovery is a solid way for us to get a feel for a project without committing to it. It is billed out separately from the rest of the project, usually at a flat rate.

### The Usual First Steps

- Kickoff meeting
- Kickoff meeting notes posted to Basecamp
- Content and UX inventory
- Strategy doc with team recommendations

### How it Works

During the team call with the client, before anything is signed, the team may suggest a separate discovery. This is a flat-rate exploration phase that allows us to analyze and provide recommendations for the project based on all information available (current desires, marketing materials, business goals, etc.).

It also allows both sides the chance to see how we work together. Plus, we have a solid idea of the scope of the whole project so we can see if nGen is a fit for both the client and the project.

Discovery is made up of three basic steps: audit, analysis, and strategy.

### Audit

The audit phase is vital to a meaningful discovery. Here we ask a lot of questions and compile all relevant data.

The basics of an audit:

- Team: **Ask** a lot of questions
- Team: **Define** and interview stakeholders (client, internal employees, customers, support) to clarify business and project goals
- Designer: **Create** a list of current brand attributes
- Team: **Identify** all available client resources and online/offline supporting materials that help define the context of the project (current app or site documentation, support resources, marketing materials, media, manuals, in-house design team, training, etc.)
- Content strategist: **Create** a document outlining current tone, voice, and core messaging
- Content strategist: **Create** a content inventory

- UX/developer: **Create** a functionality inventory
- UX/developer: **Perform** heuristic analysis
- Team: **Review** analytics of current site or app and other relevant statistics
- Team: **Assess** current and future project resources, including the client's team availability and involvement
- Developer: **Determine** if an app is required and if so, general expectations for functionality, integration, marketing and API specifications
- Developer: **Explore** what kind of hosting needs the client has based on the projected user traffic, server and security requirements, and budget
- Developer: **Determine** whether the project will need to be set on a specific framework or CMS, existing platforms, or integration with other services

## Analysis

After compiling our data, we need to organize it and create meaningful patterns.

What's included in an analysis:

- UX, Designer, Content strategist: Do an analysis of client's current branding and identity, user flow, tone, voice, and messaging
- Conduct a competitive analysis of the above
- Conduct user testing, whenever possible, to determine both the positives and the pitfalls of current product or service

## Strategy

Once we've organized the information, it's time to create a document outlining recommendations for how we plan to follow through.

What's included in our strategy document:

- A breakdown of each area that we'd like to focus on with recommendations
- Content strategist, Designer, UX, Dev: Competitive gaps; areas where competition kicks ass or falls short
- Ways we can make the client fresh: new approaches, design elements, technologies
- Plan for implementation (rollout strategy) with rough timeline
- Any areas of concern that require more exploration or attention
- Explanation of our approach and process
- The team members who will be involved and their respective roles
- Market analysis, "straw men" (unresearched) personas
- List of components anticipated for the project (e.g. API needs, payment gateway, forms, member sign-ins)

What's NOT included:

- sitemaps (unless this has been specifically discussed and included in the scope)
- design or brand attributes
- wireframes
- content

- prototypes
- design templates
- development

Here's an example of a [Discovery proposal](#) response we've sent.

You now have the strategy and recommendations you need to make a solid estimate regarding the scope and budget. It's time for the team to present the findings to the client, led by the project manager. Once the client has had a little time to "live" with the strategy/recommendation and if both the team and the client are feeling great about moving forward, a quick follow-up meeting to discuss implementation and next steps will help keep things moving. Be sure to get approval of any scope defining documents such as a list of the expected components because the client will have clarity on how additional things change the agreed upon scope.

A separate discovery is a good way to audit the types of clients nGen wants to work with.

---

**Note:** If the client agreed during the initial team call to a separate discovery—prior to nGen providing a final proposed budget—this is the time when the formal estimate and contract would be completed and presented to the client for approval.

---

## 2.2.2 Traditional Discovery

If a project is small, involves super-simple design or functionality or has a tighter budget, sometimes it doesn't make sense to do a separate discovery; a standard one will work. All the steps are the same, but the discovery phase is estimated as a portion of the entire project (usually 10-15% of a total budget, depending on the complexity and goals) and the formal documentation is more flexible. Since discovery blends into the definition phase, we want to present our understanding of the project before the definition phase begins so we all agree on what we're building. Essentially, we present our findings, estimated components needed, and recommendations for the site.

In a traditional discovery, we must informally scope the project before doing any formal exploration into the prospective client's goals and needs. This means we are presenting a range of numbers in the signed proposal which will change as our understanding of the project evolves. But we always retain the ability to go back and re-estimate when there are any changes to what is outlined during discovery and this clause is included in the proposal. We typically aim for a separate discovery since it allows us to know the ins and outs before we get in too far.

If there are any concerns (especially with regard to what is covered within the scope), we work towards a strategy internally before presenting it to the client to prevent any broken promises or scope issues. Once we've completed and presented our discovery recommendations, be sure to get written approval from the client. After discovery is complete, our team should have a better idea of the overall scope of the project, and can reference these documents to manage scope creep. Any requests that are brought in after this phase are considered work outside of the initial scope and require an additional estimate.

WAIT! SIGN-OFF REQUIRED!

---

**Note:** Regardless of the type of Discovery done, the client will be sent a [Basecamp message](#) each Monday updating them on the progress of the phase.

---

## 2.3 Phase Two: Definition

Contents:

- Phase Summary
- Typical Outcomes
  - Define The Project
    - \* Architecture Brain Dump
  - Double Check
  - Deliver the Goods

### 2.3.1 Phase Summary

- Define the Project
- Double Check Internally
- Deliver the Goods

### 2.3.2 Typical Outcomes

We try to limit any unnecessary deliverables because ultimately they take a lot of time to polish and then are thrown away. Think carefully about what the project needs to be successful and use that as the truth stick for measuring which deliverables you want to go ahead and create as a team.

- Sitemap
- Page description diagrams when necessary
- Content strategy documents: audit, analysis, and recommendations
- Content competitive analysis
- Voice and tone guidelines
- Copy; including app store content if needed
- Mental models (user flows, and current and future system support)
- User stories (narrative descriptions of how viewers currently do things)
- Updated components list; ‘statement of work’ document outlining scope
- Marketing documentation for app submission or marketing site/sign up page

#### Define The Project

Time to get to it. The definition phase makes use of all of the data we’ve collected and analyzed. Using our understanding of project goals and our detailed research, we create the skeleton of the project while defining which areas need special focus. Remember, we don’t want to predefine the project, or box it into something that already exists to make it easier to digest. Sometimes the best ideas are those that seem crazy or vague at first. For larger projects or multiple products, be sure to establish a clear timeline and rough ship dates. Keep the boat moving, phase the project—ship early and often.

The team works together to create necessary reference documents. The goal is to refine the high-level functionality, brand attributes, core messaging, user behavior, and organization of the site or app as well as its intended goals without spending unnecessary time on documentation. Building apps will involve a lot more user testing and behavior exploration, the amount dependent upon the UX goals for the project. It’s also important to remember to keep the client’s vision in mind. The product ultimately belongs to the person or people hiring us. Telling the client’s product



‘story’ back to them can keep them feeling good about the direction we’re going, not to mention reducing any scope creep down the line.

These may include any of the following (but remember, only if absolutely necessary):

- **Team:** Develop information architecture
- **Designer:** Develop a sitemap
- **Team:** Create page description diagrams to illustrate messaging hierarchy and page types
- **Team:** Assess target audience, define site goals (make sure they’re aligned with business goals)
- **Content strategist:** Complete a content document with content types, core messages, and final copy; pay special attention to process flow: things like cross linking, calls to action, images, and video placement
- **Content strategist:** Create formatting document for content; style guide, voice and tone, and nomenclature
- **Content strategist and PM:** Work with client to [create content for app store submission](#) if needed
- **Designer:** Create a mood board highlighting current and desired branding elements
- **UX:** Create usage scenarios and mental models for complex functionalities
- **UX:** Document user/process/task flow; determine how you want users to behave
- **UX:** Document interface specifications (description of UI, flow, and behaviors)
- **UX:** If there’s an open-ended budget, UX works through exploratory ideals
- **UX:** Implement usability testing with validated or exploratory benchmarks if necessary
- **Project Manager:** Outline with team all components and functionality required
- **Team:** If building an app, determine sprint and approval schedule, update project management tool with expected timelines
- **Developer:** Identify all the potential API endpoints that will need to be created in order to support the desired application functionality.

Check out the [Roles & Responsibilities](#) page to see how our tasks break down.

### Architecture Brain Dump

Designers, developers and other teammates along with other stakeholders meet and discuss how the data model and data relationships need to be represented. This helps create a shared picture of the system we’re building together.

The team fills in any gaps in understanding by reaching out to the client with follow-up questions and working internally to keep the plan flexible enough to accommodate new information. Scope creep is most likely to happen during Definition and Development. Make sure the goals and the plan to implement those goals don’t get slippery. Regular communication and documenting ALL additional requests and changes with your team and the client will help you stick to what you’ve determined fits the project budget and timeline.

---

**Note:** A good way to discuss potential scope creep with clients is to say: “That’s a great idea. Let’s look at how much additional time that will take and how much that will cost.”

---

### Double Check

Once the documents are created and polished, the team does an internal review before presenting them to the client. This is a great time to reassess high-level functionality, overall branding and tone, the intended project goals, and

make sure everything is consistent and supports the client's business goals. Feel free to reach out to one or two people outside of the team for a fresh perspective and reality check on your rationale.

## Deliver the Goods

Get it all in Basecamp and have the project manager connect with the client after submitting the definition documents to walk him or her through them. This way, any miscommunications or possible scope issues can be squared away quickly. Remember to record all additional requests, and get estimates and approvals whenever clients ask for new or different things.

The team completes two revisions to any client-facing formal documentation and any tweaks required and waits for sign-off before starting the design phase.

WAIT! SIGN-OFF REQUIRED!

---

**Note:** The client will be sent a Basecamp message each Monday regarding the progress of the Definition phase.

---

## 2.4 Phase Three: Design

Contents:

- Phase Summary
- Typical Outcomes
- Branding & Identity
  - Go Abstract
  - Conceptualize
  - Execute
- Visual Design
  - UX Design
    - \* Naming Conventions
    - \* Review Internally
    - \* Show it Off
  - UI Design
    - \* Typography
    - \* Grid
    - \* Design Components & Style Guide
    - \* Apply the Design
    - \* Flow the Content
    - \* Review Internally
    - \* Show It Off
- Application Design
  - UX Design
  - UI Design
    - \* Style Guide
    - \* Design Remaining UI Elements
    - \* Get Everyone on Board
    - \* Keep it Fluid

### 2.4.1 Phase Summary

- Branding & Identity
- Visual Design
- Application Design

### 2.4.2 Typical Outcomes

- Living style guide built in HTML/CSS
  - Including: color palettes, typographic headline, paragraph samples, buttons styles, grids, form styles, etc.
- Thumbnails and sketches
- Branding/logo files in required formats
- Design reviews and approvals
- Low-fidelity prototype/grey box wireframe
- High-fidelity prototype (IOS/Visual Design)

---

**Note:** Every designer has a process that works for him or her. You should find a process that works for you, but don't be afraid to try new things especially when those things could make you a stronger designer.

---

### 2.4.3 Branding & Identity

Polishing or redefining a client's brand and identity is about creating a visual language. Once the brand attributes have been defined and the company goals clarified, it's time to experiment with visual metaphors that reinforce the brand.

#### Go Abstract

Mini-exercises and multiple iterations keep the design loose and fluid, letting us play with and refine our ideas — including the things that feel clunky or unrelatable at first.

#### Conceptualize

Once we've got a direction, we develop some workable concepts. Then we mix our wordmarks, icons, and monograms together and distill the best ones. Filter and repeat. Keep the team involved early on and get feedback from them throughout the process. If the team is feeling good about our concepts, we show some low-fidelity sketches or thumbnails to the client. The designer has the client pick a direction based on the types of sketches that feel right so we can continue iterating.

#### Execute

At this point, most of the heavy lifting is done. It's time to define some important details:

- Typography
- Color palettes
- Icons/patterns/imagery

- Any needed print materials
- Other online materials

Illustrate how the design (logo/wordmark) will look in different environments like on t-shirts, stationery, vehicles, etc. Be sure to test how the design renders with different screen resolutions.

Allow time to get internal approval from the team for your best one or two concepts. A good critique means focusing on what's working and what's not and why. Refine the designs if needed.

The team schedules a call and the PM and designer meet with the client to review and walk them through the designs so they don't misinterpret the recommendation. Also post the design with a clear rationale in Basecamp for the client to access. Use clear communication with clients, referring back to the project goals outlined in the kickoff meeting. Ask clients to provide feedback based on what is and what's not working with the design. Gently guide them away from providing solutions for what's not working; if they'll clearly relay the issue, we'll provide the solution. This improves the ability to act on the input and allows the client to respond pragmatically. Record any additional requests by the client so the team can track them and budget for them, if needed.

---

**Note:** Refer to the signed agreement regarding how many rounds of design revisions were approved. If additional rounds are requested by the client, have the team provide an estimate and receive approval before proceeding. This should also be noted in the Monday client update as "New Requests."

---

WAIT! SIGN-OFF REQUIRED!

When the client has approved the design in writing, develop the branding guidelines and a style guide for the client. Create a folder and put design files in there. They'll get uploaded to Basecamp before the end of the project for the client's use.

## 2.4.4 Visual Design

To dive into visual design, make sure the team has completed these things:

- Research on client, their site or app, and competitors
- Defined the problems/tasks that need TLC
- Client assets, branding guidelines, etc. are available
- Content is defined, organized, written/edited (AKA no major changes will affect page layouts or hierarchies)
- Rough site map

## UX Design

Start by roughly sketching out major pages and layouts. Work closely with our content strategist to lay out a low-fidelity prototype referencing actual or virtually similar content. That way, we're not just making a bunch of grey lines symbolizing 'content'—we're actually building around the expected amount, type, and organization of content, without actually getting distracted by it. Another way to reference this type of design is to create a "live" grey box wireframe that shows the depth of functionality and lets us flesh out the major components of the site or app without boxing us into defined visuals. We're using [Travis Gertz's Live Wires prototyping method](#) which basically allows us to build a live HTML/CSS frame that we use throughout the project. It allows us to apply our design straight to the code without throwing anything away.

---

**Note:** Here are some of the basic differences between wireframes and prototypes:

---

- **Traditional Wireframes** allow you to set up your page layouts and hierarchies, but you need to create copies for each page type and links, buttons, etc. are inactive. Basically you're creating an 'image of your site or app'. Clients can't see how the functionality works together or how big the entire site or app really is.
- **Low-fidelity prototypes** allow you to navigate through the entire app or site to see all the components without seeing a defined look and feel. Makes it easier for clients to focus on the moving parts instead of what they might look like.
- **High-fidelity prototypes** is like your front-facing interface without a database or CMS hooked in (too detailed to show a client until everyone determines an early prototype direction). Once the look and feel and low-fidelity prototypes are approved, styling and the reveal of actual content in the design turns this into a high-fidelity version which is still flexible but ready for design approval before development.

We let our content, sitemaps, and layouts stay fluid during prototyping to prevent any design decisions from being set in stone. We iterate and stay experimental by alternating between in-browser designing and sketching with good old-fashioned pens. The trick is to stay as low-fidelity as possible, using grey lines in place of actual text to keep the client focused on the overall hierarchy and flow of information rather than the look and feel at this point. We also write our prototype so that we can use the same code throughout the project.

### Naming Conventions

We're working to standardize our naming conventions in our HTML/CSS code so that it will translate if/when a mobile site needs to evolve (often with a wrapper like Phone Gap or other technology that makes a site into a non-native app). We may be able to save time by having consistencies in the way we refer to elements, preventing the naming conventions for mobile and sites from clashing when translating from CSS to Cocoa.

### Review Internally

Present a low-fidelity prototype to the team. Be sure to get the feedback of any UXers, developers and content strategist to make sure everyone is aligned with core components and the scope is the same. Bring another designer in for additional feedback. Give the team enough time to provide feedback internally before showing anything to the client.

### Show it Off

Now the designer and the project manager present the prototype to the client in person or over a screen share. Walk clients through the project, explain the purpose of the prototype and feedback we're looking for so they can share the right information with their organization and feel comfortable navigating the site or app. Refer to the signed agreement regarding how many rounds of design revisions were approved. If additional rounds are requested by the client, have the team provide an estimate and get approval before proceeding. This should also be noted in the Monday client update as "New Requests."

---

**Note:** Cross-reference the documents created in the definition stage with the client while presenting the designs. Things like the sitemap, components list, and content docs will give them a clearer picture of how things fit together. Minimal changes to the site design are fine, but anything that results in changes to the sitemap, page layout, or page hierarchy needs to be noted for estimating and the requests added to the Monday client update as "New Requests."

---

Ask clients to provide written feedback in Basecamp based on **what's working for and against the design**. Gently guide them away from providing solutions for what's not working; if they clearly relay the issue, we'll provide the solution.

WAIT! SIGN-OFF REQUIRED!

Once the team gets written approval, you can start developing the style guide and cementing the look and feel.

## UI Design

The first step in UI design is for us to gather any branding guidelines & visual assets from the client and make an inventory of constraints to guide our design.

### Typography

Next, we set base typographic decisions in the browser based on client guidelines. [Typecast](#) is a fantastic tool for making and comparing typographic decisions in the browser. It's a good idea to begin with the idea of the perfect paragraph...

- Set the font
- Size
- Line-height
- Maximum line length
- Color

We can use these as the basis for all other typographic and layout decisions. This will make sure we create a tight, typographically sound design where elements are tied to each other re: size and proportion. Here's a [quick reference to basic typography](#).

### Grid

Next, we design a grid that will work across all viewport sizes and provide a robust framework for our content. Decide on a modular scale to reference to assist with determining proportional sizes for headlines, grid/gutter units, layout elements, spacing, etc.

---

**Note:** [Tim Brown's video on his Modular Scale generator](#) is an invaluable tool for creating custom scales outside of the traditional 8,10,12,14,18,21,24,48 scale. [24 Ways](#) presents another article on scales.

---

### Design Components & Style Guide

Instead of thinking in page templates or sections, instead we begin designing individual components for the app or site in HTML & CSS using [an HTML style guide](#) or something like [Brad Frost's PatternLab](#). We document small things like type styles, form fields, buttons, links, images, then move up to combinations of elements like forms, callouts, tables, and repeating content modules.

A detailed style guide is a great preliminary design deliverable we present to our clients. It lets them see how we want to move forward with the look and feel, and it clarifies how elements work with each other before cementing anything. It also gives clients a tool they can reference in the future when they want to update their site or app. The best thing is that it lets them provide early design feedback so we don't have to back up when we apply these attributes to the prototype.

[Brad Frost's Atomic Web Design](#) is a great reference for this type of approach.

## Apply the Design

After the style guide is approved by the client, we apply the CSS written for the style guide to the Live Wires prototype to begin designing in-browser. Most of our design decisions are laid out and we spend the rest of our time fine-tuning our design and fleshing out important details, page by page.

## Flow the Content

Our content strategist applies the text in basic HTML to the high-fidelity prototype while our designer applies the style guide attributes to the rest of the design. A content first approach means that we're not 'selling' content development. It's simply part of the design process. When clients provide the content themselves, we work with an experienced copywriter supplied by the client to make sure we're aligned on our content and design goals. Lorem ipsum is a fairweather friend and will not inform how much content will actually need to go on the pages. Avoid it, except maybe in places where microcopy show us when additional copy needs to be made.

## Review Internally

Present a low-fidelity prototype to the team. Be sure to get the feedback of any UXers, developers and content strategist to make sure everyone is aligned with core components and the scope is the same. Give them enough time to provide feedback internally before showing anything to the client.

---

**Note:** We usually design out a few core pages with the anticipated look and feel and get ready to show it off internally and then to the client in our first round of design. This prevents us from going too far in a particular design direction and getting feedback that forces us to change direction, eating up our budget. It also gives us a chance to review our available budget early on and re-estimate if necessary.

---

It's also a good idea, no, a GREAT one for designers to review their code before any formal QA process. Be sure to cover this time when estimating for the scope.

## Show It Off

**Round One:** It's easier to do a live walkthrough of the design with clients, then the project manager and designer follow up with screenshots of specific pages at different screen sizes (we usually stick to desktop and mobile for round one) and a written design rationale afterwards. This removes the environmental inconsistencies of the client viewing with different browser types, versions, and seeing bugs since we don't do browser testing on a moving design target (so many visual elements are apt to change). Make sure to explain the reason for the screenshots, include our Project Manager so he or she can guide the walkthrough and the designer is there to answer more technical or creative questions.

Ask clients to provide feedback based on what is and what's not working with the design. Gently guide them away from providing solutions for what's not working; if they'll clearly relay the issue, we'll provide the solution. This improves the ability to act on the input and allows the client to respond pragmatically. Record any additional requests by the client so the team can track them and budget for them, if needed. Clients provide written feedback in Basecamp so we can either re-estimate new requests or adjust our design direction and resubmit.

**Round Two:** Now that we've adjusted our designs and made sure we have enough time estimated for round two, we flesh out the rest of the pages and send our final round of designs in screenshot form to the client and get written feedback and approval in Basecamp. Additional revisions can be estimated when the client would like to see more rounds of changes.

We try to work closely with the content strategist, user experience team, and developer to support the content, user flow, and functionality of our designs.

---

**Note:** Refer to the signed agreement regarding how many rounds of design revisions were approved. If additional rounds are requested by the client, have the team provide an estimate and receive approval before proceeding. This should also be noted in the Monday client update as “New Requests.”

---

Once we receive written approval, we tidy up the front-end code, apply any last tweaks, and package up our files for development. We schedule an internal development meeting to discuss database, CMS needs, as well as any other quirks or areas that need extra love and attention. We’re now ready for development.

## 2.4.5 Application Design

App design is its own beast. If the team decides that an app needs to be created (mobile/tablet/desktop, native or not), we work in predetermined sprints to complete the work. The design and development phase are often blurry since designers and developers are working in tandem throughout the process. When in doubt about how to move forward, we talk to our teammates.

### UX Design

First, the designer or UXer outlines a user flow. This can be shared in a few different forms:

- Thumbnail sketches of screens and where each call to action leads us.
- Photoshop doc that highlights this user flow in more detail.

OR

- The biggest time saver: [Live Wires](#) ‘grey box wireframe’. Interactions and user flow are retained and the code can be saved and the design can be applied on top. Nothing is wasted.

Then the designer works rapidly with the team to outline a high fidelity prototype that will guide development work. App design differs in that the prototype can be created before the visual design since we create and apply many of the design elements while our developers complete vertically sliced sprints. Once each ‘slice’ of the prototype has been completed and approved internally, the team determines which slices the client will see. Early feedback is a must since it helps us avoid scope creep.

The team decides how best to present the design to the client and reinforces the initial business and project goals before showing each iteration. Repeating the clients words back to them helps us make sure that we’re interpreting their feedback correctly. Again, we ask them to focus on what’s working and what’s not working rather than what they like or dislike. This keeps the feedback objective and allows us to maintain our role as the experts creating their product while still being responsive to any changes the project needs to be on target.

Be sure to have the client write down ALL design feedback each time, so we can determine any additional requests, functionality, or changes in direction before they slip into the design without re-estimation. Re-estimate any changes to scope immediately.

---

**Note:** Vertical slices: an approach where an individual component of an app is completed during a sprint before moving onto another component (e.g. sign-in screens are developed, then we move onto dashboards).

---

- When developing assets, make sure to accommodate retina display first and then scale down the display to accommodate lower resolutions. Don’t forget the mobile assets. We’ll need to create high- and low-resolution versions for each. Get started early and work closely with the iOS developer to make sure the designs align with development goals and timelines.
- Make sure the team is clear about gestures and interactions. Since these differ from mobile and tablet to desktop, we need to have clarity and approval for how we’ll interact with the app. Standards FTW!



## UI Design

Now it's time to establish the look and feel. The designer works with the team to create a striking UI. Again, there are several ways to do this, but we're moving towards designing in browser since this is the most efficient, cleanest, least wasteful approach.

### Style Guide

We start with a style guide.

- The designer creates components in Photoshop so the developer can cut them (things like buttons, UI elements).

OR

- The designer designs in browser to create a responsive HTML/CSS file that can be dropped on top of the high-fidelity prototype.

The designer shares the style guide or component document internally and brings in another designer for feedback before arranging a meeting with the PM to share with the client. Try to keep the visual design revisions to two—even though we're sprinting in development, we need to keep the visual design approval process tight. This prevents scope creep during development after we apply the design. Get written sign off from the client in Basecamp. We're now free to create the rest of the UI elements.

### Design Remaining UI Elements

After getting approval for the style guide/components document, the designer codes out or designs the remaining UI elements so that the developers can plunk the design over top of the prototype. Once this has been done, it's a good idea for the entire team to sync up and make sure there are no code conflicts or UI elements that conflict with the interactions or code. Once the design has been applied and the whole team is feeling good about it, show the client a walkthrough of the design and get final approval. Again, keep rounds of revision for design to **two**. Remember, we can differentiate between the look and feel and the interaction so if the client asks for any changes that aren't related to the aesthetics, add these requests in as scope changes or ongoing sprints.

---

**Note:** It's a good idea, no, a GREAT one for designers to review their code before any formal QA process. When in doubt, use our internal QA checklist to guide your work. Be sure to cover this time when estimating for the scope.

---

Reassess the timeline with the client regularly so everyone is clear on the expected delivery. It's also a good idea for the PM to connect with the content strategist, developer, and client to begin to finalize things like required app store accounts, billing, content and keyword writing well before the app launches.

### Get Everyone on Board

Keeping all key players involved from the outset is important. Rapid iterations with the client are valuable, but the feedback from everyone on the team in-between ensures that there will be no scope creep, miscommunications, or info gaps. It's vital to do user and ongoing internal testing on the proposed app or site to make sure things work the way everyone anticipates.

### Keep it Fluid

Approvals and sign-offs for apps during this phase are obviously more fluid than they are with traditional sites. As long as there is clarity about how ongoing iterations affect the scope and budget, the team is free to create as many as the project needs to nail it down within the scope. That said, a clear schedule and related estimates for releases

and updates should be implemented as soon as possible to avoid scope creep. It's also a good idea to schedule regular product reviews with the team to manage goals and expectations.

WAIT! SIGN-OFF REQUIRED!

---

**Note:** Sign-off is especially important for app development! Making changes to a working prototype once it's been finalized creates scope creep and can keep the team designing and developing for the same release for a long time, not to mention causing the client's budget to balloon, leaving both the team and the client frustrated. Schedule adjustments in later releases to keep things tight.

---

Once the team has approved the UI design/development and they get written approval from the client, it's time for another round of testing and a final QA. Remember, development is often running in vertical slices at the same time, so lots of team planning is necessary to keep these releases in sync.

## 2.5 Phase Four: Development

Contents:

- Phase Summary
- Typical Outcomes
- Confirm Specifications
  - Review The Components
  - Pre-Production Meeting
  - API Specification
- Develop the CMS
- Finalize integrations
  - Running in Sprints
  - Continuous Testing
- Update the GitHub Repo
- Going To Market

### 2.5.1 Phase Summary

- Confirm the specifications
- Develop the CMS
- Finalize integrations
- Continuous testing
- Update the GitHub Repo
- Going to Market

### 2.5.2 Typical Outcomes

- Pre-production meeting
- CMS or database development
- Content flowing

- Brand and content guidelines

How the prototyping and production phase unrolls depends on whether the team is building an app or a site. Building apps usually involves determining more complex functionality; rapid prototyping allows the project or product to form quickly with ongoing collaboration from the team and client before development. Simple sites don't usually require development of vertical slices unless their functionality would benefit from it. They usually move from design directly into development, although we don't want to box ourselves into calling this either a waterfall or agile method. We just take the approach that works the best with each individual product or site. Design may run parallel to, staggered with, or ahead of development, and can follow up with styling CSS if necessary.

---

**Note:** For larger projects and applications, it's a good idea to have an early conversation with the client so they can establish a team to carry on support and maintenance (or hire a developer) if required.

---

## 2.5.3 Confirm Specifications

### Review The Components

It's development time. The first step is to make sure there are no gaps in information regarding functionality and its interaction with the look and feel of the site or app we are building. Ongoing communication between the designer, developer, and all other teammates prevent misinterpretation about how we translate the prototype into a functional site or application.

Before any development work is completed, we re-review our components list together and cross-check it against what we've been building so far. This allows us to see any gaps or directional shifts taken since we started and also keeps us on target for our initial release.

### Pre-Production Meeting

Once we've met with our team to go over the core components list, it's time for a pre-production meeting. Our PM and developer(s) coordinate a pre-production meeting with the designer and content strategist to determine any special areas of focus or development considerations. This is also a good way to ensure that our design build and content hierarchy are completely in sync with the development plan. Be sure we have access to all the right login credentials before any code is committed.

### API Specification

An API allows a web browser or mobile device to interact with the system to fetch or update data. Our developers work together to create documentation that outlines exactly how that API should work with the app and the steps we need to take to ensure the interactions are seamless. We also make sure we have any required account information if that API already exists or the specifications we need to create and refine APIs for our project. API building takes time and should be fleshed out as early as possible to account for all the moving parts.

## 2.5.4 Develop the CMS

Next the developer considers the CMS. If we're building an app we have to determine what, if any type of CMS, is necessary and what kind of flexibility the client should have to be able to update the content. This also means we should have a clear idea of how we're approaching the development: a CMS like [ExpressionEngine](#) or [Craft](#), or a custom built framework like [Django](#). Usually the team will adjust content for the client when building apps, since content changes affect the design of the app in later releases. For simple sites, we almost always develop the CMS that the HTML/CSS or face of the site or app hooks into.

## 2.5.5 Finalize integrations

When developing, we also need to be sure we have fleshed out all the angles of integration:

- Are we linking or using social media with the site or app?
- Are there other systems or other apps that either pull data from or provide data to our app or site?
- How much control do users have to modify this setup?
- Do we have to adjust how this process works in the future? How often?
- Are there any costs associated with these integrations (aka third party permissions/adjustments)? We have to make sure our clients are aware as early as possible.

For more traditional sites, this process happens after design has been approved. For app development or sites with databases or complex functionality, we continue to develop in vertical slices as each section of the app or site is completed, working in tandem with our designers.

## Running in Sprints

When working in vertical slices we usually coordinate one or two week sprints where we outline what portion to accomplish in that time and set clear deliverables and milestones to keep us on track.

## Continuous Testing

One of our earmarks for development has been the introduction of continuous integrative testing. Our developers create scripts that test our code, a process called ‘test driven development’, to make sure that our code ‘passes’ and acts the way it should. Ongoing testing saves time in QA since we know that the majority of the moving parts are already working as they should. We’re using a Jenkins server as a testing environment for our apps and more complex sites.

## 2.5.6 Update the GitHub Repo

If we’ve created code snippets that we’d reuse or that could be viable for the public and do not give away client trade secrets, consider adding them to a public repository on the nGen Works GitHub account. This helps us give back to the creative community but also allows us to share our process in a way that can help strengthen the industry’s design approach.

## 2.5.7 Going To Market

If we haven’t already, we make sure to discuss the desired outcome of the app with the client. Do this early and confirm often to make sure the goals of the product are still in line with our client’s vision.

- Is it going in the app store, or will it live on its own server?
- How often will it need to be updated?
- Who will be assigned to do those updates? Now? Six months from now?
- If this is a purchasable product, how does the payment structure work?
- Will there be a sales team/support team available? How will support be handled?
- How will we know when server upgrades need to happen to accommodate growth?
- Have we discussed a maintenance plan with the client? What are the expectations?

Getting a head start on these questions means we're anticipating any surprises that often tend to show up in the later phases of development as timelines and budgets run tight and clients are looking forward to seeing their babies go live. The more clear we are from the outset, the more relaxed we can be when we push the project live.

When the team is sure that all scoped work has been completed, it's time to make sure our code and design files are organized and tight and ready for a final QA. Make sure to cross-reference our component list and to fill in the client about any unforeseen adjustments in Basecamp along the way. Now we're ready for a thorough QA before launch.

## 2.6 Phase Five: Quality Assurance

Contents:

- Phase Summary
- Typical Outcomes
- Test the Waters
- Record the Bugs

### 2.6.1 Phase Summary

- Testing the Waters
- Record the Bugs

### 2.6.2 Typical Outcomes

- Quality assurance
- Browser testing
- Bug fixing

As mentioned in the Development section, our developers do integrative testing during dev to save time in QA. Additionally, our QA team, [QCat](#), tests apps at various intervals during dev to find and report issues early.

Once final development is complete, it's time for a full site or app nitpick.

### 2.6.3 Test the Waters

First the team reviews the site or app internally and tries to break it. Yes, you read that right ... we try to break the site in every way possible.

The following is just a portion of what we're testing:

- **The Look**
  - Visual & type elements
  - Spacing, headings & hover states
  - Logos, favicons, image rendering
- **The Functionality**
  - Phone, tablet, desktop layouts

- Responsive content breaks

- **The Forms**

- Error messages
- Functionality of mandatory fields
- Forms submissions

- **The Interactions**

- Page loads, buttons, live links
- Print styles, media, downloads
- Search & 404's

Next, the assigned testing team will complete a thorough audit of the entire project's design elements, content, and functionality in different browsers to ensure that everything is up to snuff. (The browsers to test have been predetermined in the proposal.)

The testing team also makes sure that no details have been forgotten, including CMS accounts, analytics, and licensing of the CMS or purchasing of add-ons used.

## 2.6.4 Record the Bugs

The testing team records any bugs in a to-do list in Basecamp so the developer can fix them once testing is complete. We include screen shots or screencasts of issues that are easier to see than explain in words. After fixes are made by the developer the item gets retested to make sure the fix didn't break something else.

The testing team has access to the CMS so can make any content-related fixes without the developer having to be involved.

When testing and quality assurance is over, and everyone is feeling like the site or app is ready, it's time to hand over the site or app to the client for a final review. Depending on complexity, plan ahead to allow the client at least one to two weeks to kick the tires prior to launch.

## 2.7 Phase Six: Delivery & Launch

Contents:

- Phase Summary
- Typical Outcomes
- The Meeting
- Presenting The Site
- Training

### 2.7.1 Phase Summary

- The Meeting
- Presenting the Site
- Training

## 2.7.2 Typical Outcomes

- Presentation of site
- Final tweaks to design and functionality
- Site or app launch
- Print materials, branding documents
- Content maintenance plan

The nitpicks are over, the team is happy, and the app or site is looking great. It's time to hand it off to the client for the final review before launch.

## 2.7.3 The Meeting

The team schedules a handoff and training meeting with the client. It's a good idea to have the whole team present for the final meeting so nothing important is left out. If the budget doesn't allow for it, make sure any recommendations or important information is relayed internally so the project manager can inform the client.

## 2.7.4 Presenting The Site

The client is provided a temporary link to the site or app where it is being hosted on a testing server. The project manager and developer walks the client through the site, the functionality, and the CMS.

## 2.7.5 Training

Our PM and developer provide the client with CMS training. The client then spends a few days reviewing the site or app and recording any bugs or errors before providing written sign-off for launch. We make sure the client knows how to record bugs properly in Basecamp so we see them and can flatten them.

WAIT! SIGN-OFF REQUIRED!

Once the client sends final feedback, tweaks have been applied to the site, and payments have been squared away, the site is ready for launch. Pull the trigger. And breathe a collective sigh of relief.

- Developer: points the site or app to the proper domain; finalizes hosting details
- Designer: uploads branding documents and style guide

We stay tuned for any bugs that might only show up when the site or app goes live. They're sneaky like that. nGen Works is responsible for any bugs or errors. Anything that breaks as a result of the client working on the site is billed out separately.

This is also a time when UX may want to conduct beta user testing with a random or controlled group of users before launching to the public. The team should agree ahead of time what type of testing should take place and for how long. UX can then measure and summarize the findings, then provide the client a report with recommendations on how to approach them.

A maintenance agreement can be finalized around the same time if the site is more complex or it's an app that needs regular releases.

## 2.8 Ongoing: Maintenance

Contents:

- [Phase Summary](#)
- [Typical Outcomes](#)
- [Site Maintenance](#)
- [App Maintenance](#)
- [Ongoing Testing](#)

### 2.8.1 Phase Summary

- Site maintenance
- App maintenance
- Ongoing testing

### 2.8.2 Typical Outcomes

- Maintenance form
- Maintenance agreement

Maintenance varies for sites versus apps.

### 2.8.3 Site Maintenance

Site maintenance is usually less intensive and can be dealt with once the client compiles a list of things that require attention. Sometimes a maintenance agreement is implemented.

Lead the client to the [QCat maintenance form](#). Once a maintenance request is received, we send the information to the team for estimating. After the estimate is approved, the work is scheduled with the team (designer or developer). Requests estimated under \$500 will be invoiced once the work is completed. Requests estimated over \$500 will require a deposit payment before the work begins, with final payment due upon completion.

### 2.8.4 App Maintenance

Because of constant updates and new releases that apps require, a solid maintenance plan is needed to maximize the efficiency of the releases. Otherwise, the team is roped into working with the client like an in-house team, which is fine if that's the plan. But if it's not, we can help the client find a replacement developer and be available for additional support when they need it.

Coordinate the details with the rest of the team and be sure to get it all in writing before the launch if possible. Follow up along the way as each respective phase is rolled out.

### 2.8.5 Ongoing Testing

We also offer ongoing testing options through [QCat](#) for sites that need browser testing or larger, more complex sites and apps that will be releasing new versions fairly regularly.



## 2.9 Glossary

Contents:

- nGeneer
- Friendgeneer
- Jellyfish teams
- Proposal
- Scope change
- Scope creep
- Team lead

### 2.9.1 nGeneer

- What we call any core team member at nGen Works. Team members with specific commitments to projects and to nGen Works are able to work on internal projects and help guide where nGen will go next. They get first dibs on all incoming projects and one core team member must be on every project.

### 2.9.2 Friendgeneer

- Specially skilled contractors that are part of nGen's network of trusted designers, content strategists, developers, project managers, etc. They are pivotal to the success of the Jellyfish Model which involves flat teams working flexibly with core team members together on short-term projects. Friendgeneers are brought in to provide additional support for big or complex projects, or to help us with our resources availability. We love them.

### 2.9.3 Jellyfish teams

- A distributed team temporarily connected by a common vision: a group of people who are accountable to each other to get things done. They create their own system, refine their roles, and create great work while in a transparent environment.

### 2.9.4 Proposal

- A legal document outlining the process, people, and general components of the project, as well as rough ballpark of the expected budget. This document can also stand as the contract. Should be vetted by your lawyer to protect both you and the client! Include a Terms of Service Agreement whenever possible.

### 2.9.5 Scope change

- A scope change is an agreed upon adjustment to the components, functionality, or complexity of an app or site that was initially approved in the proposal and that requires re-estimation. Scope changes need to be documented and approved by the client as soon as possible to prevent scope creep.

### 2.9.6 Scope creep

- Any additions, substitutions, or deletions to the original scope documented initially in the discovery phase and again during the definition phase that have not been estimated or paid for.

- Scope creep can be held in check by recording all requests and including a mandatory estimate.
- Do not add additional scope once development has begun or you'll delay your timeline and blow your budget.

## 2.9.7 Team lead

A role we introduced to make sure that someone always had an eye on the internal workings of the project and could offer support and a push to teammates.

- Supporting and leading the team (making sure everyone has what they need to succeed)
- Can be anyone on the team who feels confident
- Oversees quality of the design and/or code
- Ensures all internal deadlines and deliverables for product or project are met
- Monitoring and/or involvement in discovery, definition, design, development phase
- Attention to budget
- Attention to milestones
- Attention to scope

## 2.10 Roles & Responsibilities

Contents:

- Operations
- Business Development
- Administrative Support
- Project Management
- Team Lead
- Content Strategy
- UX
- Design
- Development

### 2.10.1 Operations

- High-level company view
- Company vision
- Tracking cashflow and company health
- Payroll support and processing
- Expense payment
- Hiring and firing
- Expensive parties with Kanye
- ...

## 2.10.2 Business Development

- Initial contact with business leads
- Sussing out business leads
- Vetting successful opportunities
- Recording and tracking all lead information
- Stats on lead success
- Proposal development
- Estimating assistance when needed

## 2.10.3 Administrative Support

- Aids in scoping and estimating
- Setting up/closing projects in Harvest and Basecamp, inviting people in, Welcome messages
- Weekly updates
- nGeneer and Friendgeneer ‘where’s the bathroom’ support
- Vacation tracking
- Invoicing
- Payments
- Attention to budget
- Attention to milestones
- Attention to scope
- Monitoring and involvement in discovery, definition, design, development, QA, launch phase
- Participation in kickoff meeting
- Tracking content and assets received from client
- Flowing content
- QA testing and control
- Staying up to speed in Basecamp throughout the project

## 2.10.4 Project Management

- Day-to-day client communication; liaison between clients and nGeneers
- Managing approvals
- Attention to budget
- Tracking client content and assets
- Vacation tracking
- Creation of and attention to milestones
- Maintaining a high-level view of the project

- Monitoring and involvement in discovery, definition, design, development, QA, launch phase
- Running kickoff meeting
- Organizing timelines and client deliverables
- Ensuring scope is well defined and clearly communicated
- Final decisions related to budget/timeline/scope

### 2.10.5 Team Lead

- Supporting and leading the team (making sure everyone has what they need to succeed)
- Can be anyone on the team who feels confident
- Oversees quality of the design and/or code
- Ensures all internal deadlines and deliverables for product or project are met
- Monitoring and/or involvement in discovery, definition, design, development phase
- Attention to budget
- Attention to milestones
- Attention to scope

### 2.10.6 Content Strategy

- Participation in kickoff meeting
- Aids in creation of planning docs (may include IA, sitemap, wireframes, PDDs, etc.)
- Verifying all planning docs meet requirements set out in kickoff and proposal
- Monitoring and involvement in definition phase, design, development, QA phase
- Auditing existing content
- Making core messaging
- Creating tone/voice guidelines
- Creating strategy for content implementation
- Creation of content
- Aiding client in creating content maintenance plan
- Creating editorial calendar when required
- Monitoring and/or involvement in QA phase
- Content flowing when required
- Attention to budget
- Attention to milestones
- Attention to scope
- Staying up to speed in Basecamp throughout the project

### 2.10.7 UX

- Participation in kickoff meeting
- Aids in creation of planning docs (may include sitemap, wireframes, PDDs, etc.)
- Verifying all planning docs meet requirements set out in kickoff and proposal
- Research (may include market research, user interviews, heuristic review, etc.)
- Usability testing and recommendations
- Monitoring and involvement in discovery, definition, design, development, QA, launch phase
- Attention to budget
- Attention to milestones
- Attention to scope
- Staying up to speed in Basecamp throughout the project

### 2.10.8 Design

- Participation in kickoff meeting
- Monitoring and involvement in discovery, definition, design, development phase
- Aids in creation of planning docs (may include IA, sitemap, wireframes, PDDs, etc.)
- Verifying all planning docs meet requirements set out in kickoff and proposal
- Creation of HTML style guide
- Creation of HTML and CSS skeleton
- Site or app design
- Illustration
- Branding and branding guidelines, attributes
- Checking design against all planning documents to verify all specs are executed
- Attention to budget
- Attention to milestones
- Attention to scope
- Staying up to speed in Basecamp throughout the project

### 2.10.9 Development

- Participation in kickoff meeting
- Getting thoroughly informed on all integration needs (client portals, ecommerce systems, client's database(s), etc.)
- Recommending technologies
- Monitoring and/or involvement in discovery, definition, design, development, QA, launch, maintenance
- Verifying all requirements in planning docs, kickoff and proposal are met
- Coding site, app

- Hooking in databases
- Testing code
- API specifications
- Finalizing server, hosting, maintenance needs or plan
- QA fixes and retests
- Attention to budget
- Attention to milestones
- Attention to scope
- Staying up to speed in Basecamp throughout the project

## 2.11 Other Helpful Information

Contents:

- Process Related Questions
  - New Business
    - \* Context
  - Discovery
    - \* Stakeholders
    - \* Goals
    - \* Timeline
    - \* Resources
    - \* Budget
    - \* Kickoff
  - Definition Phase
    - \* General
    - \* Admin
    - \* Content
    - \* UX
    - \* IOS
    - \* Design
    - \* Development
  - Design Phase
    - \* General
  - Development Phase
    - \* Technology
    - \* IOS
    - \* Social Stuff
    - \* Hosting
    - \* Payment Gateways
  - Testing/Launch
    - \* Admin
    - \* Content
    - \* Development
    - \* Hosting
    - \* IOS
    - \* Training
    - \* UX
  - Maintenance

## 2.11.1 Process Related Questions

### New Business

#### Context

- How did you hear about us?
- Why are you wanting to do this design/redesign?
- Talk about your available resources (content/design/hosting/staff).
- What are we responsible for creating?
- Is there anything unusual you, the client, are responsible for (eg. they have a designer or will do their own coding or build the site)?

## Discovery

### Stakeholders

- Who needs to be involved throughout the process?
- What kind of involvement? Contact details.
- Who provides approvals at sign-off?
- Who is our one point of contact?
- Who is our accounts/payment contact?

### Goals

- What do you need?
- Are you willing to take a phased approach if necessary?
- What are your project goals?
- What are your business goals? Now? In the next five years?

### Timeline

- Do you have a required launch date? If so, is it flexible?

### Resources

- What kind of people resources do you have? Are you willing to hire more if required?
- Do you have an in-house team of anyone involved in the design/marketing/branding process?

### Budget

- Do you have budget set aside for this project? Is it flexible?
- Who finalizes these details and is there anything helpful we should know about the process?
- What kind of additional documentation is required, if any?

### Kickoff

- What is the purpose of this project?
- What is the one thing we need to get right to make this a success?
- What is the one thing you're afraid that we won't get right?
- How would you describe the current site or app or product?
- Describe the people that use your site/app/product (age, gender, profession, frequency).
- Do you have analytics or statistics to measure your current site/app/product?
- What kind of information do you have about the way users access and use your site/app/product?



- Which tasks are critical to the process? What should users do?
- What makes visitors return or keep using your site/app/product?
- App: Are you interested in only iPhone? Android? Other platforms?

## Definition Phase

### General

- Can we receive copies of all print, marketing, online assets for analysis?
- What are the current mobile/other platform considerations?
- What is your current user base like?

### Admin

- Who will be on the team? Does anyone need to be brought in?
- What are the respective roles of these people? Is everyone clear on those roles?
- Who do we send invoices to?
- How often are we billing?
- Are we billing hourly or is this a flat project rate?
- Do I have all the contact details for our point of contact?
- Who will be creating the contract and proposal?
- Who will be editing these documents?

### Content

- What are your expectations regarding content?
- What is the current process for content creation?
- Who will be creating, entering, editing, and updating content now and in the future?
- Do you have someone who will have the required availability to provide this?
- What is the current tone of your content? What does the current voice sound like?
- How does this differ from how you'd like it to be perceived?
- What types of content will there be? (video, podcasts, etc.)
- Do you have or need a blog?
- What blog features do you need (commenting, user-generated content, moderating, etc.)?
- Do you require translation into multiple languages? Do you have contacts who do this?

## UX

- Do you have a group of folks who use your site or products available who could take part in user testing?
- Have you had any user testing done in the past?
- What do you know about the visitors that use your site?
- What actions would a typical user take when visiting your site?

## IOS

- What are the current mobile/other platform considerations?
- Is your desired application a native application? Responsive web application? Hybrid application?

---

**Note:** Native = iOS; Responsive Web = HTML/CSS/JS (not on Apple App Store); Hybrid = iOS + HTML/CSS/JS (on Apple App Store)

---

- Are you interested in only iPhone? Android? Other platforms?
- Is there an existing API the mobile application will interface with? If so, what response types does it support (XML, JSON)?
- Will you be utilizing location services (GPS) within your application?
- Do you have an Apple Developer account?

## Design

- What are you, the client, all about (what's your philosophy)?
- What are the attitudes or beliefs that shape your business culture?
- How do you, the client define yourself?
- What are five adjectives that best describe the company now?
- If you, the client, were represented by a character, who would it be?
- What makes you, the client, stand out from competitors? What's your edge?
- What's the intangible element you, the client, sells (emotions not facts)?
- How does the company intend or expect to evolve or grow?
- Who is the main target audience (including demographic information)?
- Who should be the main target audience?
- How do people currently perceive the brand? How should people perceive the brand?
- Why should the (potential) audience care about the client?
- What are the biggest challenges to sharing your message?
- What do you, the client, do well?
- What do you, the client, NOT do well?
- List competitive companies and brands.
- List any brands that are effective (local or national, even outside the clients industry).

- What current brand standards do we need to maintain (logo/colours etc.)?
- Will you require favicons?
- What current assets do you have (including images/video/content/typefaces/branding guides etc.)?
- What is your current user base like?

## Development

- What are your current logins for your CMS/hosting/analytics accounts?
- What technologies does your current app/site/software use?
- Can we receive copies of all print, marketing, online assets for analysis?
- Do you require hosting?
- What are your current domains? Do you require any others?

## Design Phase

### General

- Do the designs reinforce the project and organization's goals?
- Has the team discussed functionality and specific design elements together?
- Is the client aware of how many page types are needed, functionality constraints, and database integrations that might be needed?
- Have you updated the client on the scope, timeline, and any other factors that may have changed?

## Development Phase

### Technology

- Is there any existing data that needs to be imported? What kind and where is it currently stored?
- What are CMS needs/scope? (such as preview/draft/approval functionality; how many users; differing permission levels)
- Will there be integration with existing systems? If so, please provide login info to system(s).
- Will existing blog content need to be brought over? Comments as well?
- Any specific search needs?
- Do you have plans for using any new systems?
- Are there any specific browser needs?
- If the current site has a login protected area, need a login for team.
- Are there any technologies that must be used? (technical/functional requirements)
- What kind of forms will you require?
- What kind of reporting requirements do you have?
- What format will the files you provide be in?

- Do you currently have a domain name or do you need a new one?
- Are you currently using a content management system? If so, which one?
- Are you currently logging web metrics? If so, what metrics are you currently capturing? Do you currently have a search engine? If so, what type of search are you using?
- Do you require a unique 404 page?
- How will you be backing up any online user data?
- Do we have SSL certification set up?
- How much focus have you put toward online security? Data encryption?

## **IOS**

- Are you familiar with the App Store submission process? (if not, we will need to spend time walking them through it, or getting the metadata we need from them)

## **Social Stuff**

- Are social sharing links needed? On which site pages? Which ones?
- Will people require social networking sign ups (such as sign up via FB)?

## **Hosting**

- What are your hosting needs?
- Have we created a hosting agreement?

## **Payment Gateways**

- Do you require payment functionality?
- Do you need an online shop?
- Do you have a complete list of all products, descriptions, and specs?
- Do you have a particular payment gateway you're interested in?
- How much are you comfortable spending on a payment subscription?
- Will you use Paypal, credit cards, email payments, cheques, interac, all of the above?

## **Testing/Launch**

### **Admin**

- Have final payment stipulations been communicated?
- Have all third-party addons been included into the final invoice?
- Has the final invoice been sent?
- Have we received final payment prior to launch?

- Have we delivered any files or documents to aid the client with styles/guidelines/expectations?

### Content

- Has a style guide been provided to the client?
- Does the client have a plan for maintaining content?

### Development

- Have we tested in all relevant browsers?
- Have purchased and billed for EE licenses purchased (as well as add-on's, etc.)?
- Do we have a list of required staff accounts for Expression Engine?
- Have we completed training with the client?
- Have all bugs been identified and fixed?
- Do we have a server or information backup plan in place?
- Has Google analytics or other web metric measurement been installed?

### Hosting

- Have we sent your annual hosting invoice?

### IOS

- ...

### Training

- Do participants have the account and login credentials?

### UX

- Do we need a plan for follow-up testing?
- What type of testing will we be conducting (warm body vs specialized)?

### Maintenance

- What kind of requirements do you have?
- Has anything changed with regard to site or project goals, staff, updates, or other requirements?
- Is the site functioning as well as in the past?
- Are there any timeframe considerations, will there need to be a retainer?

## 2.12 Templates

Contents:

Discovery Proposal

Monday Update Email