

Discovering the App in Your IP

As trainers, we have all seen the “light come on” when a participant really gets a point from the training and we wonder if they will remember it in a few days or weeks when they encounter a need for that knowledge. We build job aids, tools and refreshers, and provide follow up coaching in person, online and on the phone with varying degrees of success. Apps and mobile phones offer a way for the participant to keep the knowledge close at hand.

Designing an app that bridges the gap between the classroom and the workplace is more than just a catchy phrase; it is the driving force behind building these apps.

The design for an app starts with defining the problem your user is trying to solve. However, before you can do that you must get to know your user!

Step One: Know Your User

Who will use your app? Before you can define the problem you are trying to solve, much less its solution, you need to know who your user is: Name, age, education, marital status, number of children, job position and responsibilities, salary, work situation, budget, and all the constraints and opportunities you can think of that exist for this person.

Create a persona that is typical of the people who attend your trainings. Do not be concerned that designing an app for a particular persona will narrow your market. Once the app is completed, others will find it useful as well. Ideally, you design your app to be useful to anyone in a similar business situation without having to go through your training.

Again, it is essential to be clear about the person you are designing the app for before defining the problem. This will help you make design decisions throughout the process.

Step Two: Define the Problem

Start with the problem your primary user is trying to solve. Make the question as specific as possible in terms of the outcomes – what is the work or task that needs to be accomplished? For example: “How do I get enough information in an interview to make a hiring recommendation?” or “How do I approach and communicate effectively with a specific person at work?” It is equally important to understand why your user does not do this anyway – what are the constraints?

Notice the models you use that really make sense to participants; there might be an app in there somewhere. Listen to the cynic in the room – this is the person who is telling you what problem needs to be solved; there might be an app in there somewhere.

It’s possible that you personally don’t need the help this app has to offer because you understand the concept well enough to implement it. So you must put yourself in the place of the people who will use the app, become aware of the real world constraints, problems and work situations they are dealing with everyday, and search for a solution for which an app is appropriate.

To clearly define the problem and create rigor in the product, you must be honest and realistic about the constraints of the user and the problems they face. Always remember to stay focused on solving the problem that your user has and don’t add any non-essential features.

Step Three: Model the Solution

Keep the app design small and simple in scope. Avoiding “feature creep” is the biggest challenge at this stage of design. Manage the size and scope of the app by focusing on your user’s constraints and restrictions – not their greatest wishes – just their basic needs. The usefulness of the app comes from giving your user a way to accomplish something they are unable to easily accomplish without it.

Consider that these steps of the process are not linear. Especially early on in the process, the distinction between the steps may blur. You will find that it’s an iterative process of getting to know your user, defining their problem, looking at their constraints and restrictions, brainstorming solutions, going back and refining the definitions, and ultimately organizing the data into useful information for the user.

Consider your user’s constraints and what solution you could provide. Test your solution against *all* of the user’s needs and constraints. Try as many approaches as you can think of, discard the ones that do not work and move on to the next. At this stage it is cheap to throw away an idea and start over. It is much more expensive to throw away an idea after you have begun programming. Admit your design flaws proudly and loudly, give others credit and make all your mistakes visible. This will help to underscore the fact that all great design is an evolutionary process.

Step Four: Functional Design

In the Functional Design stage, the only thing you will design and document are the most fundamental of the interactions. At their most simplistic level, Productivity apps will have three parts to them:

1. **Input** and ways to gather it
2. An **Engine** that processes the input or internal data
3. **Output** results and ways to display and transmit it

In this step of the process you will map the screen flows at a high level to describe the basic processes the app will accomplish without going into too much detail. For example, you can map out 10-12 potential screens on Post-it notes and arrange them in a logical flow. It's important to document how the information and the user will move through the app using the Home Screen, Input and Output Screens, and fundamentally how the Engine will work.

- **The Home Screen** is an anchor point from where the user can access the main functions of the app. Ideally it should have no more than three buttons with simple commands.
- **Input Screens** are where the user enters data, makes selections from existing data resident in the app, or otherwise provides information that will feed into the Engine to accomplish the task.
- **Output Screens** show results such as lists of information, action plans, assessments or other data that offers guidance for accomplishing or completing the task. Consider adding the ability for users to email output to themselves or others, if it makes sense.
- **Document the Engine** and roughly how it will function to process the data.

After successfully defining the problem, look into your intellectual property to find the model, structure or concept that will help deliver the solution you are after. That model, structure or concept will become the basis for the app's Engine. The Engine is the construct that helps to organize the information and present it in the prescribed manner. This can be a matrix, a hierarchical structure of information, the major steps in a process, or most any other clearly understood ways of organizing the information.

At this stage of the design you will work through the details of how the Engine actually works and learn what will be required in the way of content development later on. As a way to test your understanding at this point, write a draft of the app's description and define the content required in the next step.

Step Five: Detail Design

In the Detail Design stage every screen is defined. This means every button and potential action of the user must be detailed and considered. After each screen is drawn, insert arrows to connect the logic flow of the app. This is best laid out in the form of a grid of tiles the size of a smartphone screen leaving space in between for notes and arrows.

Navigation

The app's navigation is determined, tested and mapped in the Detail Design. Good navigation design will help the user feel they are in the driver's seat and are getting something done. They should never feel lost or uncertain about what to do next.

To help users navigate, primary navigational elements are placed at the top of the screen in what is called the Nav Bar, and secondary navigational elements are placed at the bottom of the screen in the Tab Bar.

Menu buttons on the Home screen and elsewhere should be concise and descriptive to help users identify the implications of clicking the button. As a general rule, use verb-noun pairs to name your buttons because that does an effective job of making users feel they are in command of the controls.

For example:

- Select Questions
- Create New Plan
- Access Existing Plans
- Answer Questions

Home Screen

The Home Screen is the anchor point of the app, so spend the time to get it right. When your user is here they should have a sense of comfort of having been here before. App users are extremely impatient and unwilling to put up with a bad user interface or one that leaves them feeling lost. The average app is given about 30 seconds before the user decides to keep it or throw it away. Sobering statistic. Your Home Screen should be so intuitive to use that NO ONE is ever lost at this screen and it should be immediately clear what to do first/next.

Follow Each Thread

Beginning with the Home Screen, build a flow of the app's interactions that emanate from each menu button, one screen at a time, following each interaction thread to the logical end of its flow. Follow all of the

primary interaction threads in the design to completion in this way. Explore lots of different solutions and do not be afraid to throw away something you have designed. Often the first design of a great app is awful compared to the eventual product.

Work through the Nav Bar instructions on each screen as you go. All of the major threads in the app should start at the Home Screen and eventually lead to the final Output Screen. Sometimes it may be necessary to start with the final Output Screen and work your way backwards. Once your design pieces all fit together like a puzzle, the final step is to go through and place the Tabs in the screens where appropriate. The result is a detailed chart representing the flow of all of the screens in your app.

Validate the Design with a Paper Test

At this point, conducting a paper test is the best way to ensure that you've documented every possible step and function of the app in a user-friendly way, and test the assumptions you have made about the problem you are solving for the user. The biggest challenge to the designer at this stage is to listen to the testers. Ask open-ended questions and then just listen. Do not explain, do not justify, do not rationalize. Just listen and say thank you for the valuable feedback.

Find 10 to 12 people who can assume the role of your user. Cover all of the screens on your flowchart with Post-its with the exception of the Home Screen. After all, only one screen at a time will be revealed on the smartphone so you want to emulate that. As the user touches a button, uncover the connected screen and move the post-it note to cover up where they just were, leaving only the open screen visible. Follow this process noting where they get confused or lost, buttons they do and do not select, etc.

Walk the tester through screens as you ask questions like “Which button do you want to select next?”, “Why?”, “What do you expect to see on the next screen?”, etc. If the tester looks confused, find out what's confusing them. Take notes on all their thoughts. This is how you will validate the overall flow and individual interactions designed into the app as well as gather new ideas for features to add in future revisions.

Step Six: Organize and Develop the Content

This stage includes developing all of the static and dynamic text that will appear on all sections, screens and buttons. This includes everything! The content should be helpful, brief and to the point. With app content size matters, and shorter is better.

Content is organized into the following ten areas:

1. Name of the App
2. App Description
3. Buttons
4. Nav Bar and Tab Bar
5. Dialogue Boxes
6. Tutorial
7. Assessment Questions
8. Assessment Results
9. Lists
10. General and Miscellaneous Content

Step Seven: Software Development, Alpha

Obviously one of the critical players on your team is going to be a software developer. In our case, we chose a business model of using an external partner for the technical development to allow us to focus on what we do best. This section will focus on your role during software development. There are several tasks that will fall to you depending on the development model you choose, but primarily your role ought to include content management and product testing.

Creating Alpha code is the first step in software development and the goal at this stage is the delivery of an Alpha build. The Alpha build is a software release that is “feature complete” and can be tested internally but may still have some serious bugs causing it to crash or not operate correctly.

Use internal people to do the testing at this point, as they know how the app is supposed to work and often are best at using temporary “work-arounds” to exercise the code. Here, based on your situation, you might want to test components of the app and do the system integration in the next stage.

Bug List management begins here and is maintained all the way through Release of the product. Document all bugs but at this stage focus on fixing the serious bugs that make your app crash. Bugs usually involve “interaction”, meaning the app does not act as it is supposed to, but there are also “look and feel” issues to fix and these also must be tracked. These can be just as distracting to the user as problems with the interaction, so treat all bugs as equal in the end.

Step Eight: Software Development, Beta

The goal at this stage is a Beta build, a pre-release version of the app that is given to external testers. Beta testers are a group of previously identified people who have smartphones. It is best if they are not familiar with the product to this point and ideally they represent your targeted user group. This pre-release build will have more bugs than the

eventual released product, so testers need to be people who can see their role as “finding bugs” not complaining about them. Even the relatively minor bugs must be systematically hunted down and eradicated.

There is a difference between a bug, something that does not work as it is supposed to, and a design flaw, something that you do not like the way it works. The reason that it is important to distinguish the two is that a design flaw will take more time and money to correct and a bug is usually a simpler, less expensive problem to fix. A bug can be fixed now and a design flaw might be able to wait until Version 1.1 to fix.

Testing the app is critical to your role even though the developer will have done some testing on it before they release it to you. Go back to the paper Detail Design document and test “every corner” of the design to ensure the app works as it is intended to and that it looks the way you want it to.

The adage “You can have it Good, Fast, or Cheap – pick two” applies in building apps with a slight variant. An app project usually encounters the dilemma of “Schedule, Quality, or Features – pick two”. It is possible that as you reach milestones on a specific *feature*, there will be too many difficult and unresolved bugs affecting the *quality*, and the time required to fix them will impact the *schedule*. Given the zero tolerance most users have with buggy apps, it may come down to a choice between schedules and features.

Step Nine: Polishing

In the Polishing stage your time and attention should be on the tiniest of details. No issue is too small to address. Spelling is important and critical at this point! Anything that is distracting should be examined – a font size that is too small, something that is not the right color, screen movement that is slugging, etc. This is the last chance to get it right.

Regardless of the platform or app store you choose for the app, having an accurate app description is necessary to the Release process. This is the time to revisit the app description for clarity and message and ensure all the features are accurately portrayed.

At this stage, if the smallest thing is out of place it can keep the app from being approved and released. Details, details, details.

Step Ten: Release

Once you have built your app, tested it, and have everything working to your satisfaction, you can submit it to the App Store. If it is an iPhone app, Apple will test it for bugs, memory leaks, etc., and anything they consider to be objectionable content. Apps have also been rejected for duplicating functionality already on the iPhone.

Apple’s basic intention is to keep the quality high as well as protect the user from any poorly running or potentially harmful apps from being distributed through the App Store.

In the case of our first app, The Interviewer, it was about two weeks from submission to approval. In the case of The Influencer approval was granted in a matter of days.

In the Future

As technology does its continuing march in support of corporate learning and development, and as mobile devices become more ubiquitous and powerful, apps will play an ever increasing role in Performance Support, Coaching, Mentoring and Talent Development.

How do you get started? As with any new initiative, you start by doing your homework – by understanding the habits, constraints and current technology of your existing population and the opportunities that this offers. Recognize that no matter what you build today will most likely be obsolete in three years so start soon!

By 2012, the Mayans predict the world, as we know it will end and Forrester estimates 73% of the workforce will be mobile, are you ready?

Accelerate Learning & Development, Inc.

5321 Scotts Valley Drive, Suite 208
Scotts Valley, CA 95066 USA

Phone: (831) 438-8500

Email: no-pressure@accelerate-ld.com

Website: <http://www.accelerate-ld.com>
