



# Project Complexity Worksheet

## The Goal of this Work Sheet

The goal of this work sheet is to provide a starting point to define and quantify the type and scope of work involved with your specific project. Each area has it's own level of complexity and expertise required for implementation.

The Classifications are broken down into the following areas:

- Platform(s) : iOS, Android, WebServer, Cloud, etc. This relates the number of platforms that are involved in the project. Just because you want a mobile app does not mean that you may need server resources.
- Data Integration : Local-FS, Local-DB, Web, WebServices, etc. This relates to the data used, generated or access via the mobile app.
- Data Types : Data Entry (text), internal media, external media, images, video, location, etc. This relates to the type of data used by the project.
- UI Complexity and Features : This defines the type of UI implemented in the application along with the type of features that are contained within.
- Developer Experience : This section talks about the level of experience in specific areas of development. As a projects scope increases the amount of difficulty can increase and overwhelm the original project team. This section contains a list of items to watch out for.

## Does the Project Need to Exist?

Everyone comes up with a good idea at some point, but that does not mean that it should be made into an application. As a software engineer I've been approached a number of times by friends, neighbors and acquaintances on an application that "I" should write. Easy for them to say. In developing software, time is money and it's my time. The exception to this rule is I love to help people to learn about software development, what the complexities and issues are and what they can expect when looking for a developer for their project. That is what this document is hoping to provide.

Back to the question of "Does the project need to exist?" If there are multiple related Apps available then it may not make sense in developing yet another one. The exception are 1) if the existing apps are not professional, 2) missing features or 3) you can provide added values by integrating other services, etc. If the App is to support an existing business then development costs can be viewed as part of doing business. Most Apps today are either free or at low cost to the user, so the reason to develop an App to make money may be one of the lower reasons for developing one. However, there are a number of Apps (MySines is one) that was developed to purely help users. To provide a free service to people is a good reason to develop and app, as long as you know what to expect.

# Basic Questions to Ask

These are the questions that can be asked about the idea / project that is being asked to be developed. Answers to these questions should provide a good high level outline of the App's scope and basic complexity (excluding games). You may not have answers to some of these questions and may need to think about how to answer some, but it's a start.

Here are the basic questions:

## **For a basic mobile App the questions are:**

- Does this already exist on the App Marketplace?
- What would make your App different / better?
- Are you expecting to make money on this App?
- Does the App augment an existing business?
- Do you have funding in place for development?
- Do you want to support both iOS and Android?
- Do you want a tablet version?
- What screen sizes do you need to support?
- Do you have documentation for what you want?
- Is this a Game?
- Are there any licensing / copyright issues?
- Do you have graphics already?
- Will you be collecting personal information?
- Will you be dealing with financial information?
- Are there any background processes that need to occur in the app?

## **For an App that accesses a WebServer:**

- Will it be accessing your WebServer (vs general Servers).
- Are there any licensing / copyright issues with accessing the WebServer or its data?
- Are there any privacy / financial information that will be sent / retrieved from the server?
- Will you be providing other parties with information contained / collected by the App?
- Will you be collecting information from multiple specific servers?
- Are there any time critical issues with the servers (response / request within  $x < n$  seconds, etc.)?

# Factors Effecting Development

## The App User Interface:

Because mobile devices have limited visual real estate, each screen can only contain so much information, thus requiring the App to be broken up into operational sections that transition between each other. We can refer to these operational sections as views and the interaction between them and other supported views (camera, maps, etc.) as the workflow of the App.

In the worksheet we will break down the different areas of the UI (User Interface).

## Data

The amount, type and location of data that an App works with greatly effects the complexity and cost of developing the App. Simple App's like a ToDo list is minimal to implement but the same App being able to sync with other devices running the same App becomes a much larger issue when synchronization, storage and a Webservice become a concern.

The basic levels of data are:

- Local FS : (FileSystem, preferences).
- Local-DB : (Sqlite, CoreData) that require a DB schema be designed for querying and any other related data operations.
- External : This is data that is off device and accessed via Webservice or another method. This raises the complexity for working with a different platform, language and behavior with requirements for security, scalability and reliability (backup?). These type of projects can be open ended as the scope of this work varies widely unless every aspect is known up front and that just is not going to happen.

## Experience

This aspect of estimation of the project also varies widely and has the most impact on development. It's not un-common for development of a project to be effected by a factor of 10 (either way) by the experience and quality of the people doing the development. The number and types of technology used in software development is so large that not every developer / engineer (more on this later) can know, let alone be proficient in all aspects of development. For example, we at Blue Footed Boobie do not develop games. That is a specialized field that is best handled by those that are experts in that field.

Within each field of development (App, DB, Webservises, scalability, etc). there are levels of complexity. One person that can handle Android Activity, may be in over their head when off-device synchronized data will multiple types of active states are required for the App.

For each section in the worksheet there is a field for Experience for that specific area. This may just be a matter of the developer doing a bit more learning to provide some functionality, but it is important to make note of.

The developer should already have a "skeleton" App that contains most of what is needed for most mobile App's. This provides a starting point for your App and provides a sure foundation going forward.

## WorkSheet - Platforms

Platform	Required	Dev-Units	Experience
<i>iOS - iPhone, iPod-Touch</i>			
<i>iOS - iPad</i>			
<i>iOS - base version supported (4.2+)</i>			
<i>Android - Phone (2.2+)</i>			
<i>Android - Tablet (4.0+)</i>			
<i>WebServer - UI - (client, JS, toolkit)</i>			
<i>WebServer - OS - (Windows, Linux, other)</i>			
<i>WebServices - (Rest, SOAP, other)</i>			
<i>WebServer - DB - (MySQL, Oracle, MSSQL, other)</i>			
<i>Specific Technologies (AirPlay, Printing, etc.)</i>			
<i>FaceBook, Google and other API's</i>			

## WorkSheet - Data Integration / Type

Data	Required	Dev-Units	Experience
<i>Local data with FS / Preferences</i>			
<i>Local data with DB / CoreData</i>			
<i>DB with &gt; 5 tables</i>			
<i>Images / Video in Data</i>			
<i>Streaming DB from WebServers</i>			
<i>Streaming Video / Images from WebServers</i>			
<i>Consuming WebServices (XML, JSON, other)</i>			
<i>Editing Images / Video</i>			
<i>Total size of expected data (&lt;1mb, &gt;5mb, streaming, etc.)</i>			
<i>Login / Off usernames / passwords</i>			
<i>Private and / or Financial data</i>			

# WorkSheet - UI Complexity

UI	Required	Dev-Units	Experience
<i>WorkFlow Layout (Tree, linear, other)</i>			
<i>Number of Sections (setup, main, supporting, etc)</i>			
<i>Skinning Level (none, light, heavy)</i>			
<i>Calling external views / activities (camera, maps, contacts, etc)</i>			
<i>Async operations with Background threads</i>			
<i>Amount of Data rendered (light, medium, heavy)</i>			
<i>Multiple States (Login/Off, Basic,Advanced, limited, etc.)</i>			
<i>Gestures Types used (swipes, pinch, taps, etc.)</i>			
<i>Special Views (swap, flip, scrolling, page, etc.)</i>			
<i>Primary Navigation (Tab, Split, Nav-bar, etc.)</i>			
<i>Total number of Views / Activities.</i>			
<i>Dynamic control creation based on data.</i>			

## Columns

- Required : This column is if this item is applicable.

- Dev Units : This column can be almost anything that denotes the level of complexity based on the values contained. For example : The "Total number of Views / Activities" item value can vary based on the number of pages. A 4 page App would be low -vs- an App that contains a total of 25 views. The value used is relative for all items.

- Experience : This column can be used by the developers to denote areas where further research may be required or a specific person may need to handle.

The general gist of the work sheet is to force yourself to place in writing the aspects of App to get a rough guild as to it's complexity. To a lay person, developing an App is just the creation of a few screens with fields. For a professional application the hidden scope can be much larger and that is what this worksheet is hoping to provide.