

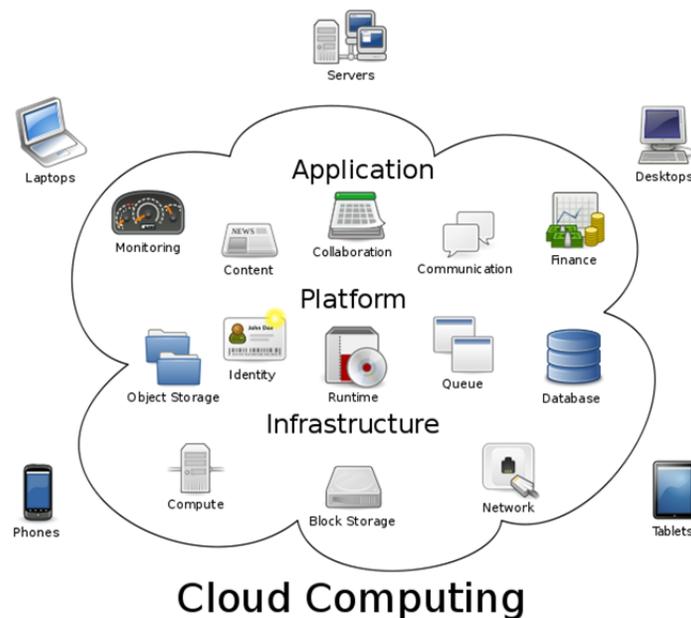
How Big is My Cloud?

By Roger Heller

When cloud computing first came about I recall talking to a colleague and remarking that the cloud is just a repackaging of timeshare. We joked and praised large mainframe manufacturers for finding a way to repackage their systems into super servers to continue to draw in the big bucks. Well that was a while back, and now, unbeknownst to me; I am hooked on the cloud. I realized that I get my music from the cloud, my books comes from the cloud, I rely on the cloud to get boarding passes for airplanes, I pay bills through the cloud, I play word games in the cloud with friends across the country and I even manage my recipes in the cloud. If I'm such a cloud user and a measurement professional, I should be able to figure out how to come up with pertinent measures that cloud developers and users can utilize to help them manage their development efforts. This article is my attempt at providing simple powerful techniques that can be used to establish the size of cloud applications.

What to Measure?

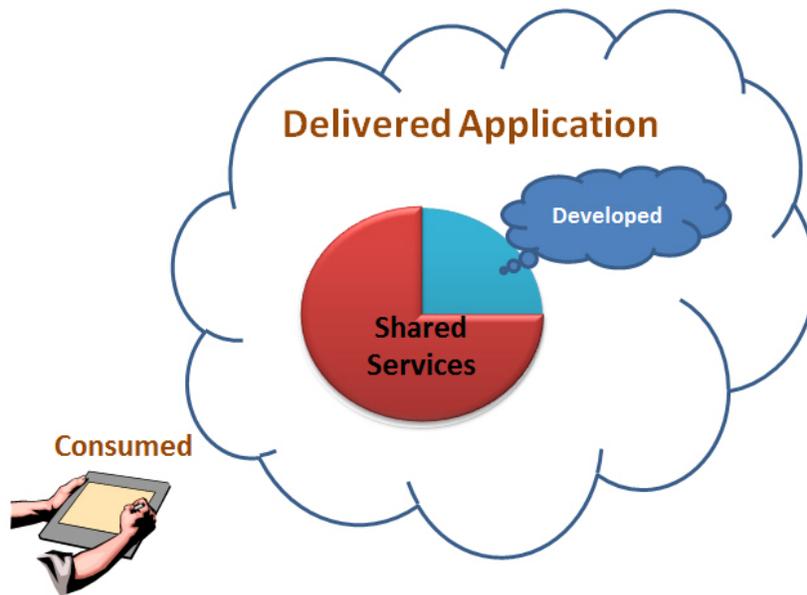
The first big challenge is to figure out what to measure. There are many moving parts to the cloud, as depicted by the diagram¹ below, and a cloud developer only has control over a small subset of these parts at any point in time.



¹ This image was obtained from Wikimedia Commons, authored by Sam Johnston on July 24, 2011.

Appropriate size measures can be developed for each piece of the cloud, but it is important to only measure what you can control. For the purposes of this article the scope of the measures to be discussed includes the Cloud Applications and the various methods that enable users to access the applications.

From one perspective there are at least three aspects of a cloud application that need to be considered for sizing. These are: Developed Functionality; Shared Services (i.e. messaging, collaboration, email, VoIP); and Consumed Functionality (functions actually used by an end user).



If the objective is to establish metrics on productivity, cost, quality and schedule as it relates to custom development, then the focus should be on the Developed Functionality. In this case, the size will only reflect those functions that are developed from scratch or customized/enhanced from existing functions. If the focus is on establishing metrics on what is potentially delivered to a user of the cloud applications, then the focus should be on measuring both the Developed Functionality and that portion of Shared Services used by the application to deliver the functionality. In this case, the size would include all of the functions, including reusable or out-of-the-box functions that needed no customization to meet the functional needs. Having this size can provide insight into the total functionality being delivered to the end user, whether it is used or not. If the focus is on what the user actually uses of the Delivered Functionality then the measures should be aimed at what is consumed. Focusing on the Consumer Perspective can help to direct future development. One approach to developing software is to be of the opinion of “if we build it they will come”; another perspective can be “if they will use it I will build it.” The truth lies somewhere in the middle and both perspectives need to be considered. So sizing a cloud application from a consumer’s perspective can give developers an idea of what consumers use and what they want, which can influence future development requirements,

estimates and costs. Measuring each or all of these depends on the purpose behind the measurements.

What to Include and What to Exclude

The first step in measuring anything is to define where to begin and where to end. When measuring an application's functional size the best way to determine the beginning and end is to pick a point of view. If the point of view is to measure a news application then the functions that are made available to the various individuals that interact with the application to provide and consume content should be considered. If the point of view is to measure a shared word game hosted in the cloud, then the functions used by the players and the functions required to support play should be considered. If you are only interested in measuring an enhancement effort rather than a total development effort then the focus would be on those functions that have been modified as a result of the enhancement project. But the point of view would still be from a user's perspective (consumers and providers), focusing on those functions that are used in the application being measured.

Now that we have discussed what functions should be counted, the next challenge is to discuss when they should be considered. There are some schools of thought that say a function is a function no matter how or where it is accessed. For cloud applications this simple view would make measuring a function easy. Just ignore all of the different platforms a user has (i.e. the various devices such as PCs, mobile phones, tablets used to access the cloud) and all of the unique development that has to take place to make each work properly and just identify a single function. But this approach is fraught with potential problems. It makes it nearly impossible to compare different development efforts where one effort may be to develop for a single platform while other development efforts may be for multiple platforms. It is very likely that the effort to develop a solution to run only on a PC is going to be less than that required to develop the solution to run on both a PC and a tablet, yet, taking the single function approach, the delivered functionality is viewed as identical. The resulting productivity, cost and quality metrics may be of little value to the organization.

The best way to avoid this dilemma is to consider, given some basic rules, that each platform is unique. With this approach, if a solution is available on multiple user devices then each should be considered to be a unique platform. For example, if a cloud application is only available on a mobile phone, then that would be the single platform and the functions would only be counted once. If, however, the application is developed for both a mobile phone and a PC, then it would be considered to have two platforms, and where reasonable unique functions would be identified. If the application was enhanced at some point to also function with a tablet, then there would be three platforms and a third set of functions could be considered.

Of course there has to be some guidelines to control the number of platforms included in the scope, but they are simple. If the solution runs on multiple platforms with no modifications or

customizations to accommodate this capability then all platforms should be considered to be the same and the functions would only be considered once. The other guideline has to do with multiple similar platforms. In this case if the application runs on multiple mobile phones providing identical functionality and the only difference is the operating system utilized by the phone then all mobile phones are considered to be a single platform. The fact that the application has to be modified to run in different operating systems is considered to be technical adaptive maintenance and any effort associated with making it work should be tracked as maintenance, not development or enhancement effort.

Likely, there are individuals who will take the perspective that developing the software to run on multiple platforms is also technical adaptive maintenance and therefore all of them together should be viewed as a single solution. More often than not, in today's environment similar end user functionality is offered on different families of platforms (i.e. mobile phones versus PC's versus tablets). But often, the functions provided while similar, can vary from one family to another. In some cases the amount of data provided is different or the functions themselves are different. Therefore, the set of functions each platform family provides should be viewed as a unique solution.

To be clear, each situation needs to be evaluated on its own merit and if making an application available on multiple families of platforms is truly a technical port, with no impact on user functionality, then all of the devices should be viewed as a single platform. Whatever the ultimate decision, it should be clearly documented within the organization to ensure that the reasoning behind the measurement direction taken is understood and agreed to by all interested parties.

How to Measure Cloud Functionality

Up to this point I've discussed just what can be measured, not the how. In fact there are probably several good techniques that people have come up with to measure cloud applications, but nothing discussed so far leans heavily towards one or the other. Given my background, the example I'm going to discuss is based on the International Function Point User's Group (IFPUG), function point analysis (FPA) technique. There are reasons as to why function points are a good technique to consider. First, it is well established. The FPA technique has been around and is known by many people and is generally accepted as an industry standard size measure. Secondly, the technique is fairly simple to apply and it doesn't require a huge learning curve. Individuals that are already trained in the IFPUG FPA methodology can easily use this knowledge to size cloud projects and applications. Thirdly, it is an international standard which can be applied to any computer based application. So, for these and other reasons, my first choice is to use IFPUG function points as the size measure for my cloud.

The subset of the cloud I'm going to size is a Recipe application. The following will attempt to classify the functions of the Recipe application based on which functions have been developed,

which functions are provided through shared services as well as functions that are consumed by the end user.

Recipe Application Description

The Recipe application provides a simple mechanism to capture Recipes either through manual entry or importing from various sources. The Recipes can be categorized based on a list that is built from scratch. Recipes can be retrieved, edited and printed via a browser on a PC. Using different downloaded applications the recipes can be accessed, but not modified, using a tablet and/or mobile phone. Each device can have concurrent access to the Recipe application. The following diagram depicts this cloud application.



The first consideration is to define the type of count to be completed. For this exercise I am going to establish the size of the Recipe application from the cook's perspective. Therefore it is a baseline count that will represent the subset of the application's functionality available to the cook. The next step is to identify the scope of the count which will focus on the end user functions provided through the Recipe application. The next activity is to identify the applications that will be included in the count. As mentioned earlier, there are many moving components in a cloud application but when looking at it from a functional perspective we can only be concerned with those pieces that are recognizable to the end user. So, while in fact the Recipe application may rely on platform as well as infrastructure capabilities to make it function, from a user's point of view the logical application boundary is simply the Recipe application and the functions that one would utilize. Of course there is the fact that there are multiple

platforms that can be utilized with the application’s functions. These include PC’s, mobile phones and tablets. And, in fact, each of these utilizes different physical applications to access the functions of the Recipe application. The PC uses a web browser; both the tablet and the mobile phone have separate and unique downloadable apps. But at the end of the day, the primary reason these physical applications exist is to support the end user functions provided through the Recipe application. Therefore, each of these physical applications along with the physical cloud applications would be logically grouped into a single boundary representing the Recipe application.

Now that the application boundary has been identified, the next step is to determine the logical functions that will be included in the count. As discussed earlier, each platform, in this case a PC, a mobile phone and a tablet, provides similar, yet different, capabilities to access the application. Therefore, the unique functions provided to the user through each different device will be considered to be transactions.

This approach is completely in compliance with the FP counting rules related to multiple media since there is unique processing logic for each and it makes sense. Effort is required to develop and maintain these different devices. Defects are going to be reported against them and costs are going to be aligned with them. These three data points (effort, defects and cost) provide the basis for measuring the productivity, cost and quality as it relates to the Recipe application. If all three devices were lumped together into one set of functions, then the resulting measures would be misleading and potentially useless.

Now that the boundaries have been identified and the different types of functions that should be included have been determined, the next step is to identify the specific logical functions that will determine the functional size. The following represents the functions available through the Recipe application.

Table 1 – Functional Description

Function Name	Function Description	Developed	Shared Services	Consumed
Create and Maintain User Profile	These functions are specific to this application and should be considered unique	Yes		Yes
Login	Allows the user to access the Recipe application upon launching. Requires User name and password.	Yes		Yes
Import Recipes	The application allows a user to import from 10 different websites by copying the URL of the Recipe and pasting into an import command. Each site has a different layout and format for the data they provide.	Yes		Yes

Function Name	Function Description	Developed	Shared Services	Consumed
Create Recipe	Allows the user to create a Recipe manually by entering in a title, description, ingredients, cooking instructions and nutritional information	Yes		Yes
Edit Recipe	Both created and imported Recipes can be edited on a PC from the Recipe detail page. Recipes can also be deleted from this page.	Yes		Yes
Share Recipe - Email	Ability to share a recipe through various social networks		Yes	No
Share Recipe - Twitter	Ability to share a recipe through various social networks		Yes	No
Share Recipe – Face Book	Ability to share a recipe through various social networks		Yes	No
Recipe list	Provides a list of all Recipes entered. A Recipe can be deleted from the Recipe list.	Yes		Yes
Recipe categories	Recipe categories can be added and deleted. They cannot be changed.	Yes		Yes
Shopping list	A shopping list can be added to and edited. The final result is printable from the PC and is maintainable and viewable on all devices.	Yes	Yes	Yes
Weekly planner	A Recipe can be added to a weekly meal planner that can be retrieved on the PC and printed and is maintainable and viewable on all devices	Yes	Yes	Yes
Print Recipes	Allows the user to print the recipes when attached to a PC		Yes	Yes
Scale Recipes	Allows the ingredients of a recipe to be increased/decreased by adjusting the number of servings. Only available on the PC	Yes		Yes
View Recipes	Recipes can be viewed on all of the devices	Yes		Yes
Share Recipes	Provides the ability to share a recipe via email or other social media means		Yes	Yes
Create Menu	Allows the user to build a menu for a particular meal by clicking on specific recipes	Yes		No

For simplicity sake this article will assume that all of the functions included in this analysis are of low complexity and the Value Adjustment Factor isn't being utilized. The appendix to this article contains a summary of the function point count of the Recipe application for the functions listed above. The chart below summarizes the major contributors to the functional size of the Recipe application.

Table 2 - Function Point Summary

Functional Component	Developed	Shared Services	Consumed
Internal Logical Files	42		35
External Interface Files			0
External Inputs	84		75
External Inquiries	42	18	45
External Outputs	4		4
Total	171	18	159

So how big is my cloud? Well it depends on how you look at it. The cloud Recipe application delivers 189 function points (171 developed + 18 shared services) to the cook. But I only consume 159 of the total that is delivered. Knowing what has been developed can be used to establish productivity, cost and quality measures. Knowing what is delivered is useful in knowing the overall size of the application and can be used for portfolio management and application planning. Knowing what is consumed can result in a more targeted development effort going forward.

The Recipe application represents 159 of my cloud function points. At last count the total size of all of the applications in my cloud was over 1,000 function points and growing. How big is your cloud? Take this technique for a spin; you might be surprised by the size of your cloud and how easy it is to figure out.

Appendix A – Recipe Application Function Point Count Example

Function Name	Function Type	Developed FPs	Shared Services FPs	Consumed FPs
User profile	ILF	7		7
Recipe file	ILF	7		7
Category file	ILF	7		7
Shopping list file	ILF	7		7
Weekly planner	ILF	7		7
Menu	ILF	7		
Create user profile	EI	3		3
Maintain user profile	EI	3		3
Retrieve user profile	EQ	3		3
Login	EQ	3		3
Import recipes	EI	3		3
Create manual recipe	EI	3		3
Edit recipe (import or manual)	EI	3		3
Delete recipe	EI	3		3
Add recipe categories	EI	3		3
Delete recipe categories	EI	3		3
Retrieve recipe category	EQ	3		3
List recipes – PC	EQ	3		3
List recipes – Mobile phone	EQ	3		3
List recipes – Tablet	EQ	3		3
View recipes details – PC	EQ	3		3

Function Name	Function Type	Developed FPs	Shared Services FPs	Consumed FPs
Scale recipe – PC only	EO	3		4
View recipes details – Mobile phone	EQ	3		3
View recipes details – Tablet	EQ	3		3
Print recipe details – PC only	EQ		3	3
Add recipe to shopping list – PC	EI	3		3
Add recipe to shopping list – Mobile Phone	EI	3		3
Add recipe to shopping list – Tablet	EI	3		3
View shopping list – PC	EQ	3		3
Print shopping list – PC	EQ		3	3
View shopping list – Mobile phone	EI	3		3
View shopping list – Tablet	EI	3		3
Edit recipe to shopping list – PC	EI	3		3
Edit recipe to shopping list – Mobile Phone	EI	3		3
Edit recipe to shopping list – Tablet	EI	3		3
Delete shopping list – PC	EI	3		3
Delete shopping list – Mobile phone	EI	3		3
Delete shopping list – Tablet	EI	3		3

Function Name	Function Type	Developed FPs	Shared Services FPs	Consumed FPs
Add recipe to weekly planner – PC	EI	3		3
Add recipe to weekly planner – Mobile phone	EI	3		3
Add recipe to weekly planner – Tablet	EI	3		3
Retrieve weekly plan – PC	EQ	3		3
Retrieve weekly plan – Mobile phone	EQ	3		3
Retrieve weekly plan – Tablet	EQ	3		3
Edit Weekly plan – PC	EI	3		3
Edit Weekly plan – Mobile phone	EI	3		3
Edit Weekly plan – Tablet	EI	3		3
Share Recipes – PC	EQ		3	
Share Recipes – Mobile phone	EQ		3	
Share Recipes – Tablet	EQ		3	
Create Menu PC only – add	EI	3		
Edit Menu PC only – change	EI	3		
Delete PC only Menu	EI	3		
Retrieve Menu	EQ	3		
Print Menu PC only	EQ		3	
Total Summary Count		171	18	159