Cost, Effort and Pricing Estimation

Explanation of Function Point Table

Number of User Inputs – 7. Account creation requires 4 inputs, 1 each for the fields of username, password, email, and phone number (even though the last two are optional). Setting a time limit also requires an input. Inviting an accountability partner can either use email or phone number, so that would add 2 inputs.

User Input is low complexity because all inputs are either just strings or numbers.

Number of User Output – 19. The main sources of user output would be the data visualization on screen, and the different files that can be exported. There are 5 different forms the data can be visualized in, and 5 different file types for export. The other outputs are prompts for user input or selection. Account creation and login both require prompts for their corresponding input. Data visualization and data export both require a prompt to select the type. A notification must be displayed prompting for an override if a time limit has been reached. Setting a time limit requires a corresponding prompt, as well as inviting an accountability partner. The menu and tutorial will each need their own output functions too.

User Output complexity is high, since the data visualization and file output are very involved and dominate the complexity, even though the other outputs are just simple prompts.

Number of User Queries – 12. Login would involve 1 query. Requesting override requires a query, as well as selecting the option to invite accountability partners. Both data visualization and data export would require 2 queries, since they need to be chosen, and then a specific option must be chosen after that. Searching for accountability partners in the app takes 1 query. Manipulating the menu requires another query, and interacting with the tutorial may involve up to 3 queries.

User Query complexity is low due to the fact that most involve simply selecting an option from a menu, although some involve searching and comparison to a database.

Number of Data Files – 3. There is very little data stored, rather than directly accessed from the phone. One file would contain user information, including the information necessary for login. One contains all the app usage data for the specific device. A third file contains data for different apps, such as what category they belong to.

Data file complexity is low, reflecting that the files are mainly lists.

Number of External Interfaces – 4. Device I/O and device background processes are the two most important. Some sort of texting and emailing interface must be used to invite accountability partners that aren't already users of the app. The usage data of all the apps on the device is the last interface.

External interface complexity is high, since mobile operating systems have some idiosyncrasies, and the application may be developed for multiple environments

	Function Category	Count	Low Complexity	Average Complexity	High Complexity	Count * Complexity
1	Number of User Input	7	**3**	4	6	7*3=21
2	Number of User Output	19	4	5	**7**	19*7=133
3	Number of User Queries	12	**3**	4	6	12*3=36
4	Number of data files and relational tables	3	**7**	10	15	3*7=21
5	Number of external interfaces	4	5	7	**10**	4*10=40

GFP = 251

PCA

Does the system require reliable backup and recovery? 4

Are data communications required? 5

Are there distributed processing functions? 5

Is performance critical? 0

Will the system run in an existing, heavily utilized environment? 4

Does the system require online data entry? 5

Does the online data entry require the input transaction to be built over multiple screens or operations? 3

Are the master files updated online? 5

Are the inputs, outputs, files, or inquiries complex? 3

Is the internal processing complex? 1

Is the code designed to be reusable? 1

Are conversion and installation included in the design? 2

Is the system designed for multiple installations in different organizations? 0

Is the application designed to facilitate change and ease of use by the user? 3

Sum * .01 = .41

PCA = .65 + .41 = 1.06

FP= GFP * PCA = 266.06

Productivity = 8.75 FP/person-weeks

This assumes 4 hours to complete a "Function Point" and 35-hour work weeks.

Effort = 266.06/8.75 = 30.41 person-weeks

Time = 30.41/4 employees = 7.6 weeks

The development proper will take 8 weeks.

These 8 weeks predicted by the Function Point estimate will take place in months 2 through 6 of development. For each month, 2 weeks were budgeted strictly for development, so the project allocates 10 weeks in total for code development. Most of the remaining time is used testing, receiving feedback, reviewing feedback, and modifying requirement specifications. The extra 2 weeks serves as a reasonable buffer in case the estimate of 8 weeks was flawed or unexpected complications occur.