



Establishing Application Boundaries

Dan Horvath

The establishment and placement of application boundaries is an important part of any function point count. All of the function point functions are affected by the placement of the application boundary. Thus the Function Point Analysis Process detailed in the Counting Practices Manual specifies that the identification of counting scope and boundaries be performed early in the process, just after determining the type of count.

Some considerations regarding the identification of application boundaries include the type of count, the user's view of the application(s), the application boundary identification process itself, and the impact on the present and future function point counts.

TYPE OF COUNT

The first step in the Function Point Analysis Process is to determine the type of count. The type of count influences the placement of the boundary because if the count is a new development count, the application boundary to contain the new functionality will need to be established or possibly changed. If the count is an enhancement count, then functions within an existing application will be affected. Assuming the application boundary has already been established, there may be no impact.

Even enhancement counts may impact the boundaries, however. For example, functionality could possibly be removed from one application and added to another. Also, even for enhancement counts there may be some new functionality. The determination of which

application the new functionality belongs to may not be trivial.

USER'S VIEW

The user's view of the application is by far the most important factor in the determination of application boundaries. Remember that the user, in this instance, is defined as the person or persons who provide and approve or business requirements for a system.

The user must be able to understand and describe an application as a single entity. This would be the case if the user, at some point, provided requirements for the establishment of the application. It would also be the case if the user understands other applications as being separate entities in terms of business function.

It is important to note that user requirements, and not technical considerations, are used to establish the boundary between applications. For example, one technical team may support back-end batch processing for payroll functions, while another team may support web front-end payroll functions that operate on a different platform. Although the technical teams and platforms may differ, if the user view is that of one set of business functions (payroll processing, in this instance), then it is *one* application boundary.

THE APPLICATION BOUNDARY IDENTIFICATION PROCESS

Once the type of count has been established, the boundaries can be identified and established. Identification means describing the application

as an entity or business function, including all functions that are part of it. Establishing the application boundary means documentation of this information. As noted above, the process is mostly based on the user's view of the application(s). There are, however, additional considerations as well.

Step 1:

The first step is to gather and examine the relevant information about the application(s). Some strongly suggested documentation includes:

- A Context Diagram
- The Entity Relationship Diagram
- Process Flow Diagrams
- The Project Charter (if applicable)
- The Requirements Specifications (if applicable)

Additional documentation that may be helpful to the identification of project boundaries, and will also likely be required for function point counts includes:

- The Business Design Specifications
- Screen Layouts
- Report Layouts
- File Specifications
- The Technical Design Specifications

Step 2:

The second step is to discuss and further explore the documentation with Subject Matter Experts and/or Users. This may be done as part of a function point counting session, or separately, prior to the actual count. If the boundaries are to be defined during the count, it is important that they are established before identifying data and transaction function types.

The Context Diagram should depict a one-page graphic of the system under consideration. It should clearly identify both new and existing

flows both into and out of the system. This visualization of the system is very important to the establishment of the boundary. If this cannot be done, the system either needs better definition, and/or it is perhaps more than one application.

The Entity Relationship Diagram (ERD) describes the data stored for a system in graphical terms. During the function point count, the ERD will be used to help identify Internal Logical Files. For the purpose of identification of boundaries, however, the ERD can help answer some questions: Is there more than one database used by the system? If so, how is the data related? If relationships between the databases are weak or non-existent, there may be more than one application boundary. Conversely, if there are several strong relationships between the data, the various entities may be part of one application boundary. Thus, the ERD helps in the decision of what data is included in what application. As an example, an ERD may show all entities required to store Call Center data. If most of the entities are related in some way, there is a good chance that the Call Center system should be considered to be a single application. A different ERD may show Payroll and Personnel data in such a way that there are few relationships between the two types of data. After also reviewing the processing functions at a high level (are the personnel and payroll entities maintained separately?), there may be an indication that in this case, Payroll and Personnel are two separate applications.

Process Flow Diagrams describe the movement of data. If data is generally moved into and out of a single location, even if in several steps, this would tend to indicate that it is one application. For example, a set of Process Flow Diagrams may show several flows into and out of a sales application. Some of the output flows may have more than one step, such as the creation of an intermediate file from which other extract files

are produced. In this case, there is still one location that is the original source, so this would tend to indicate that there is one application. More than one “central location” may indicate more than one application. Therefore, the Process Flow Diagrams help in the decision of what processes are included in what application.

For both new development as well as enhancement projects, there ought to be a Project Charter. The Charter describes the reason the project is to be conducted. For example, the reason for a project may be to enhance a billing application to add a web-enabled front-end to a batch system. Examination of this information will help define the boundary.

The Requirements Specifications describe project requirements in detail, as specified by the user. Using the Project Charter example above, the user-specified web-enabled front-end functions to be included would be clearly stated in the Requirements Specifications. As with the Project Charter, the examination of this information can help define the boundary.

Step 3:

The third step in the boundary identification process is to state and confirm the project boundaries. After exploring the documentation with the Subject Matter Experts and/or Users, it should be possible to make clear decisions identifying the application boundaries.

If not – if there are “loose ends” or conflicting directions based on the documentation, a determination must be made regarding what, if any, additional information or people are required. Perhaps the documentation was not clear or complete. Perhaps Subject Matter Experts or others disagree about the boundaries. Or perhaps the most knowledgeable people were not available as needed. In these cases, the correct information must be gathered, and/or the correct people included.

If there is still an impasse, the User(s) should decide. Remember that in the function point world, “User View” rules. In any event, once the decision has been made, the User should be able to understand, confirm and agree with the boundaries.

Step 4:

For step four, the boundaries as identified, are to be documented. This entails describing the application(s) in terms of their functionality. The description should include, at a high level, the data and processing functions that make up each application.

THE IMPACT OF APPLICATION BOUNDARIES ON FUNCTION POINT COUNTS

The placement of application boundaries does impact all function point counts against those applications. The impact may be great or small, depending on the functions themselves. Here are a few considerations:

If data from one logical file is used to populate a different file, the process may not impact the function point count at all if the two files are part of the same application, and no other data crosses the boundary. Conversely, if the two files are defined as, say, Internal Logical Files (ILFs) in two different applications, then the data transferred from one to the other can be counted. For this example, perhaps the process may be an External Inquiry to extract and send the data from one application, and an External Input to update the ILF in the second. In cases where there are many such instances of data being “sent” from one application to another, this may be an indication that the boundaries should be re-examined; there may be one application as opposed to two.

If data is maintained in one application and referenced in another, then there is an Internal Logical File in the one application, and an

External Interface File (EIF) in the application that references it. Had the two applications been identified as one, then there would only be the one ILF, with no EIF.

If data is identified as a logical file, and is maintained in two different applications, it should be counted as an Internal Logical File in each. If there are a large number of these types of files, this may be another indication that the

application boundaries should be re-examined; once again, there may be one application as opposed to two.

By considering the impact on function point counts, one can understand the importance of the correct placement of application boundaries.

About the author

Dan Horvath, Senior Management Consultant, Q/P Management Group wrote this article. Mr. Horvath specializes in project management and software engineering metrics, including function point analysis.