

# Aligning Function Point Counting and Test Planning

## 1.0. Introduction

With the ever-increasing trends to make IT departments more productive to compete with outsourcing, synergies that can be found in the organization play an important role in decreasing the overhead costs. Two sources of overhead are the cost of measurement and the cost of quality. Although both play an important role in the management and control of an IT department, many people in an organization view them as something else to do along with the project deliverables. One source of synergy is the removal of duplicate work. There are some similar fundamental activities that take place during function point counting and test planning. This paper will explain how activities can be leveraged to remove duplicate work when **aligning function point counting and test planning**.

## 2.0. Some Brief Background on the Test Process

Testing is a process to measurement software quality that is used as input to a judgment of risk when releasing a system to the user population. Aspects of a system are tested and the results are analyzed until the project group and/or the project stakeholders decide that the level of possible risk remaining in the system is acceptable to migrate it to production. A group or individual can test a system by operating it under controlled conditions and evaluating the results. There are several reasons that testing is performed. By eliminating issues, or bugs, in a system prior to its migration to production, risks can be reduced which could help to obtain the following:

- Reduction of problems occurring in production systems
- Reduction of software support costs
- Increase of development predictability
- Increase of confidence that a system is acceptable for its intended purpose
- Increase of savings for time and money

A subtle defect can have a significant impact. For example, it would likely make a significant impact if an application used an incorrect algorithm to calculate monthly interest charges for customer accounts. Either the company could lose revenue by not charging enough, or lose customers by over charging. By efficiently adding quality to an organization, it will lead to a competitive advantage.

Some components to a solid test process include:

- Test Plan
- Test Cases and Scripts
- Test Traceability

The test plan defines the test scope and objectives, as well as the strategy that will be used. This document generally remains as a high-level document. The test cases and scripts give the details on how each component of the system can be tested. The test traceability documentation allows the ability to determine what components have been tested, what have not, which have passed, and which have failed.

### 3.0. Where is the Duplicate Work

In my experience as a software tester, I've seen three separate decompositions of a system for three different purposes. In this case, the obvious one would be for function point counting. The other two are for work breakdown structures and test planning. By educating test planners with function point knowledge, they can structure their test traceability from requirements to tests so it includes transactions used for function point counting. Testers would make good candidates for counting training, because they are familiar with breaking down systems into testable processes. They also have an objective view with the user activities in mind. Not every tester will be able to effectively learn how to count. If they can, they will be a valuable project resource.

Most of the information needed for both testing and counting are the same. They are just used for different reasons. During the creation of tests, the test planner has insight to the information used for function point counting, but they generally do not understand the function point twist on those concepts. The following table helps to illustrate the gap.

Function Point Lingo	Tester Lingo
Internal Logical Files (ILFs)	Target tables for data validation testing
External Interface Files (EIFs)	Referenced or source tables for data validation testing
Record Element Types (RETs)	Data dependencies
File Types Referenced (FTRs)	Data sources
Data Element Types (DETs)	Data fields used (unaware of triggers and messages)
External Inputs (EIs)	Data maintenance transactions
External Outputs (EOs)	Reporting or control population
External Inquiries (EQs)	Reporting or control population

Providing adequate training has taken place, it would take little added effort to collect this information along the way. The function point counting meeting at the end of the project could be greatly expedited by having most of the system mapped out for the group to review.

Additionally, test execution can be mapped to function points. For example, metrics could be created showing function points tested, function points passed, function points failed, etc.

### 4.0. How to Approach Alignment

There are two testing concepts that I highly recommend. The first is documenting the test traceability of your projects. This concept involves listing out all of the system requirements, and mapping which tests prove which requirements. In some cases, multiple requirements may be tested by one test script. In many cases, it will take multiple test scripts to test a single requirement. The second concept is "smart numbering" your test scripts. One method is to categorize each script with a Class and a Procedure. An example of such a breakdown is contained in Appendix A. This allows testers to know the nature of a test script just by looking at the name. By combining these concepts a matrix can be created.

For example, assume that you have a system requirement that the user should be able to enter, edit, and delete a standard order and a replenishment order. I realize that this is an over-simplification, but it is just to illustrate a point.

The matrix for this may look something like this:

Business Req. #	Business Req. Name	Functional Req. #	Functional Req. Name	Test Script
BR 01	Create Order	FR 01.01	Standard Order	02.201.01 – Create Standard Order
		FR 01.02	Replenishment Order	02.201.02 – Create Replenishment Order
...	...	...	...	...

Additional, but similar, rows would be added for editing and deleting.

The appearance of the matrix can vary from company to company or even project to project. The most important thing is that it fits the situation. If there are Use Cases, add those to the matrix. This way we can tell that if script 02.201.02 fails, then there is a problem with BR 01, FR 01.02 (a.k.a. creating a replenishment order).

Some things that will be tested can be identified as elementary processes in the function point counting world, and some may not. In the above example, the creation of the two different order types could be two separate elementary processes due to the different data required for each. In some cases, scripts will need to be written to check for things like navigation or performance. These are not elementary processes. With the addition of some columns, function point counting information can be collected along the way while planning for the system testing.

## 5.0. Example

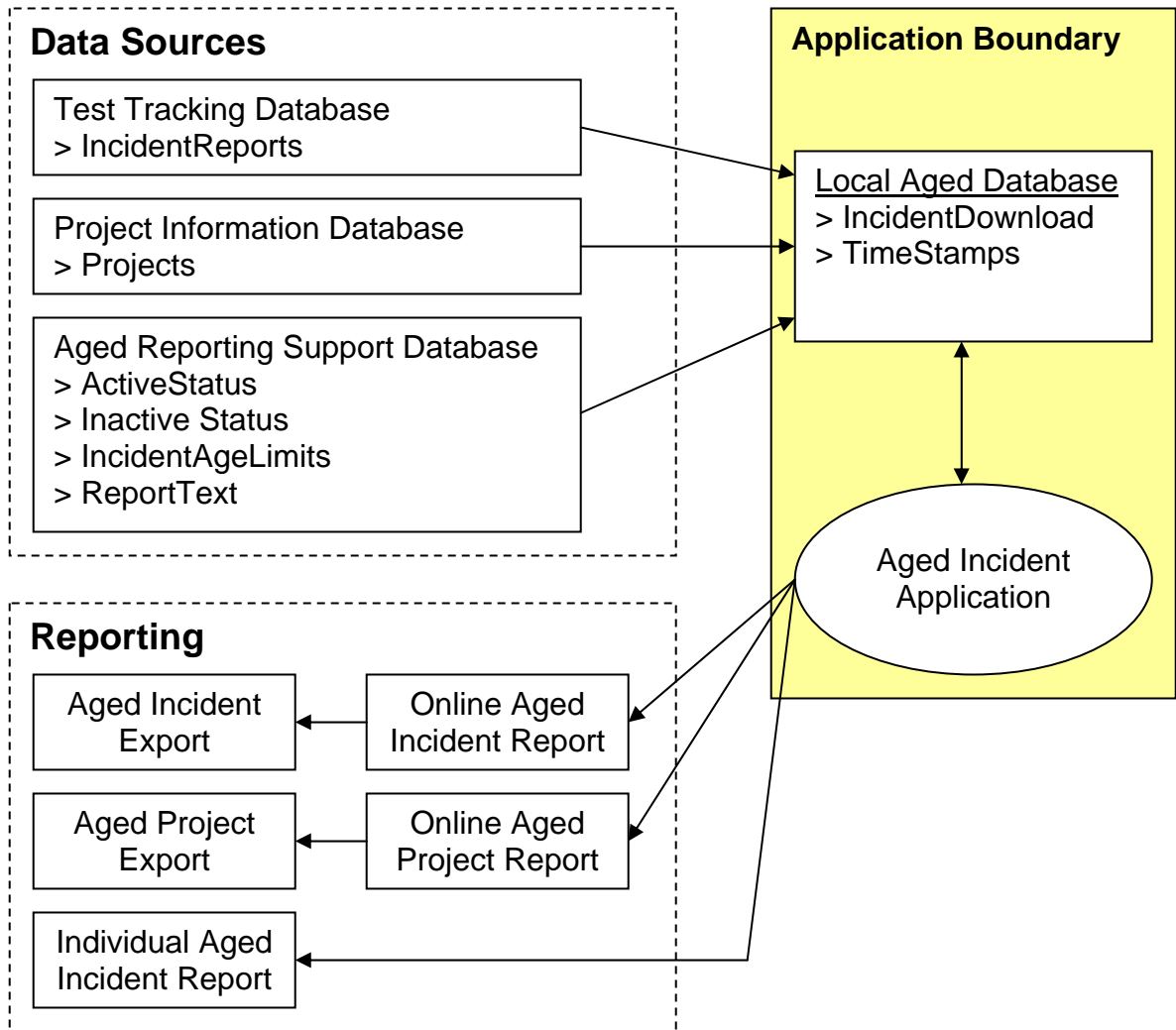
I will now walk through an example using an application that creates questionably aged incident reports for a test-tracking database based on some business rules. Since the GUI exists, I will use it to help make things a little easier to follow. Appendix B contains the Business and Functional Requirements (BR and FR, respectively) that were used to create this application.

### 5.1. Boundary Decisions

When planning to test a system, it is important to understand what the system looks like. In making decisions on the test points, questions need to be answered for items such as:

- What information is flowing into the system, and what is its source?
- What functions do the system perform, and what is effected by it?
- What kind of output will come from the system?

In some cases, an overall diagram of the system may not exist, or does not represent a functional view of the system. In this situation, the test planner needs to diagram the system from a testing (or functional) point of view. The following diagram shows the basic system flow, and a decision made for the application boundary. The application boundary decision was made from reviewing the inputs and outputs of the system.



## 5.2. Data

The first step to understanding how to test a system is to understand its infrastructure. To be able to perform data validation testing, the data sources need to be identified. This information can be obtained from data maps, physical data models, and discussions with members of the development team. Appendix C contains a copy of the data tables used for this application.

Required information for testing purposes includes:

- Where is the data located?
- How is the data loaded?
- Are there interdependencies involving the data sources?
- What is the data used for?

Some of these questions should ring a "Function Point Counting" bell. Some of the questions are answered below:

- The Aged Reporting Support Database is a centralized database referenced by each user's local database. It contains tables (**ActiveStatus**, **InactiveStatus**, **IncidentAgeLimits**, and **ReportText**) that are referenced to process data as it is written to the local database or reports. Data contained in this database is not maintained by another application. It is contained in the database to make it easier to query against, and to keep it external to the application so a change will not involve a recompile. This data is important to test planning, because a tester will know that the maintenance of the data does not need to be tested, just referencing it does. (To a function point counter, the fact that the data is not maintained in any application means it is neither an ILF nor an EIF.)
- The Project Information Database is somewhat similar to the Aged Reporting Support Database. It is a centralized database referenced by each user's local database. It contains several tables, including the pertinent **Projects** table, which is referenced to process data as it is written to the local database. Data in this table is maintained by a data maintenance application used by members of the testing group. Again, this data is important to test planning, because a tester will know that the maintenance of the data does not need to be tested, just referencing it does. (This one, however, is maintained by another application, so a function point counter would realize that this is likely an EIF for this system.)
- The Test Tracking Database is a centralized database, which is again referenced by each user's local database. It is a non-relational database that contains a form for recording **IncidentReports**. Data is maintained by a database in the Lotus Notes application, so testing the maintenance is not required, just referencing it is. (Again, to a function point counter, this is likely another EIF for this system.)
- The Local Aged Database is on each user's machine. It holds data for the user so they may maintain and edit the records, and not change another user's work. The first table is the **IncidentDownload** table. There is a process to import data from the Test Tracking Database into the local database. Each field brought over will be used for calculations or reporting. Additional fields on each record will be used to associate the Test Lead; calculate age and associate any violations; and to maintain who should be informed of that incident. The other table is the **TimeStamp** table. It holds specific events and when they happened, so the system can be managed. It contains the date data was downloaded, which is also displayed to the user. This information tells the tester that the processes to populate the two tables need to be tested to validate the data. (To a function point counter, this identifies two probable ILFs.)

So far, this is information that would need to be collected to plan for testing. With the extra function point counting knowledge, it could be determined that there are two EIFs and two ILFs. If the knowledge of function point rules is there, the data used for counting can be collected along the way.

Information Needed for Testing		Shared Needs for Function Point Counting	
Data Sources	Data Validation Testing	ILF/EIF	RETs / DETs
ActiveStatus	No	---	--- / ---
InactiveStatus	No	---	--- / ---
IncidentAgeLimits	No	---	--- / ---
ReportText	No	---	--- / ---
Projects	No	EIF	1 / 3
IncidentReports	No	EIF	1 / 13
IncidentDownload	Yes	ILF	1 / 21
TimeStamps	Yes	ILF	1 / 2

### 5.3. Transactions

Test scripts walk the tester through the system's transactions, to verify that the reactions are expected for each action taken by the tester. The tests to run are determined by project walkthroughs, requirement documentation, use cases, or any other source of information that can be obtained about the system. The ideal mantra for a tester should be, "Leave no stone unturned." Even if something is not tested for some reason, you want to know that it exists.

The requirements (Appendix B) are a good place to start determining what items there are to test. Appendix D contains the GUI windows for additional reference.

#### 5.3.1. Business Requirement 01

##### BR 01 Background

BR 01 addresses obtaining the data from external sources and compiling it in a local database along with calculations and derived data. This information will be placed on the **IncidentDownload** table. All of the Functional Requirements under Business Requirement 01 happen at the same time, so they would likely be tested using the same script. There are no choices to be made during the processing, so it is done the same each time. Function Requirement 01.06 states that the Aged Incident Grid should be displayed after the data is loaded. This will be handled separately from the data load process, because this report can be triggered separately from the data download. It is just part of the data load flow, because it assumes that is what the user will want to see first. It is performed to save a step.

##### BR 01 Analysis

The tester will have to understand the calculations and conditions encapsulated in the business rules to make sure that all the data conditions are represented in the test data. This analysis will allow the tester to fully understand what is happening during the data download. Many data sources are touched during this process. **ActiveStatus**, **InactiveStatus**, and **IncidentAgeLimits** are referenced to get information for calculations and derivations. **Projects** and **IncidentReports** supply base information for the new records. **IncidentDownload** and **TimeStamps** are both updated as a result of this process. Fields, calculations, and data derivations would be identified in preparation for test queries to be constructed. The data download process is initiated by clicking the [Get Data from Lotus Notes] button, and the main visual result is a date/time display in the updated time text box (See Figure D1).

The traceability matrix for BR 01 would look like this:

Bus. Req.	Fun. Req.	Script
BR 01	FR 01	04.403.01 – Import incident data
	FR 02	
	FR 03	
	FR 04	
	FR 05	
	FR 06	(See BR 03)

**What a Tester would see:**

- There are 3 reference sources (**ActiveStatus**, **InactiveStatus**, and **IncidentAgeLimits**)
- There are 2 data sources (**IncidentReports** and **Projects**)
- There are 2 data targets (**IncidentDownload** and **TimeStamps**)
- There are 13 fields being pulled from the **IncidentReports** data source
- There are two associated fields on the **Projects** data source, and 1 separate field is being pulled from it
- The 3 reference sources will be queried to determine 4 fields involving age and any violations
- There are 3 fields are defaulted to 'No' for sending personalized incident reports to specific people
- A field in the **TimeStamps** data target will be updated with the date and time
- A text box on the GUI will be updated with the date and time
- A button click triggers this activity

**What a Counter would see:**

- There is one elementary process
- The primary intent is to maintain two ILFs (**IncidentDownload** and **TimeStamps**), so it is an EI
- There are two EIFs (**IncidentReports** and **Projects**) involved, so the FTR count totals 4
- There are 14 DETs being updated on the **IncidentDownload** ILF from the two EIFs
- There is 1 DET being updated on the **TimeStamps** ILF
- There are no different DETs referenced on the EIFs
- The button to initiate the process is also counted as a DET, so the DET count totals 16

**Summation**

- **EI with 4 FTRs and 16 DETs giving it a High complexity level and 6 Function Points**

**5.3.2. Business Requirement 02**

**BR 02 Background**

BR 02 addresses the fact that the user should be able to pull up a list of the aging rules at any time while using the application.

### BR 02 Analysis

This requirement is fairly simple. A button, [Aging Rules] was created outside the tab structure so it will always be visible (See Figure D1). When the user clicks on it, a new window opens displaying the aging rules. These rules are contained in the data source **IncidentAgeLimits**. Some data testing will have to take place, since the rules are contained in a database for ease of maintenance. Since there are no ILFs or EIFs referenced, this will not be counted. It is merely navigation.

The traceability matrix for BR 02 would look like this:

Bus. Req.	Fun. Req.	Script
BR 02	N/A	04.403.02 – Display Aging Rules

#### What a Tester would see:

- The rules are contained in a database, so a query will have to be written to test the content in the application
- The tables the data come from are not maintained by an application
- A button that must always be visible opens a separate window to display the information

#### What a Counter would see:

- There is nothing to count here, because the data sources are not ILFs or EIFs. The information could be hard coded and get the same effect. This is navigation.

### 5.3.3. Business Requirement 03

#### **BR 03 Background**

BR 03 addresses the Questionably Aged Incident Report. The user can generate the full report by clicking on the [Reload Questionably Aged Incident] button on the “List of Items” tab (See Figure D1). The user can also generate a filtered report by making selections on the “Filter The Incidents” tab and clicking the [Update Questionably Aged Incident List] button (See Figures D2A and D2B). The window will be redirected back to the “List of Items” tab to display the report. The report can then be exported to MS Word by clicking the [Export to Word] button (See Figure D1). The export to MS Word is not always done, and does not contain the record count display. It is considered a separate process. Appendix E contains a sample of the incident export.

#### **BR 03 Analysis**

The report is only one report; it can just be filtered using input from the user. The report can either be executed by clicking the [Reload Questionably Aged Incidents] button or the [Update Questionably Aged Incident List] button. The general filter (See Figure D2A) applies a name condition to the Created By, Test Lead, and Assigned To attributes of the record. The detailed filter (See Figure D2B) applies the filters individually, so there is no need to count them separate. There is a calculation involved to determine the record count, so it can be displayed to the user.



The traceability matrix for BR 03 would look like this:

Bus. Req.	Fun. Req.	Script
BR 03	FR 01	02.205.01 – Aged Incident (Unfiltered)
	FR 02	02.205.02 – Aged Incident (General Filter)
	FR 03	02.205.03 – Aged Incident (Detailed Filter)
	FR 04	05.503.01 – Export Incident Report to MS Word

**What a Tester would see:**

- The report can be run from the “List of Items” tab without filters
- The report can be run from the “Filter The Incidents” tab with filters
- The general filter allows the user to select a name for the Creator, Test Lead, and Assigned To; which allows them to return any associated records with that person
- The detailed filter allows the user to specify a Project ID, Creator, Test Lead, and/or Assigned To; which allows them to return a more specific set of records
- The user can decide which of the 11 fields they want to show, by checkbox designation
- A record count will be displayed for the report
- The process is initiated by clicking a button
- The same report that is displayed (without the record count) can be generated in MS Word
- All data comes from the **IncidentDownload** data source

**What a Counter would see:**

- There are two elementary processes: one to create the online report, and one to create the MS Word report
- The user can provide input for the process by designating Yes/No to display each possible field, 11 DETs
- The user can provide filter values for the report for 4 fields, but they are already shown on the report and can't be counted twice
- Both the online and MS Word report has 11 standard pieces of repeatable information, 11 DETs
- The online report has a record count 1 DET, but the MS Word report does not
- A button click initiates each report, 1 DET
- There is one FTR, **IncidentDownload**
- The primary intent of both processes is to present information
- The online report has a calculation, so it is an EO
- The MS Word report does not have a calculation, so it is an EQ

**Summation**

- **EO with 1 FTR and 24 DETs giving it an Average complexity level and 5 Function Points**
- **EQ with 1 FTR and 23 DETs giving it an Average complexity level and 4 Function Points**

### 5.3.4. Business Requirement 04

#### BR 04 Background

BR 04 addresses the Questionably Aged Projects Report. The user can generate the report by clicking on the [Load Questionably Aged Projects] button on the "List of Items" tab (See Figure D1). The report can then be exported to MS Word by clicking the [Export to Word] button (See Figure D1). The export to MS Word is not always done, and does not contain the record count display. It is considered a separate process. Appendix F contains a sample of the project export.

#### BR 04 Analysis

The report can only be executed by clicking the [Load Questionably Aged Projects] button. A calculation is performed for this report to determine how many days ago the final incident was closed, and then determines that it was over 60 days to display it on the report. All data is coming from the **IncidentDownload** table.

The traceability matrix for BR 04 would look like this:

Bus. Req.	Fun. Req.	Script
BR 04	FR 01	02.205.04 – Aged Projects
	FR 02	
	FR 03	05.503.02 – Export Project Report to MS Word

#### What a Tester would see:

- The report can be run from the "List of Items" tab
- The report shows 4 pieces of information for each record
- A record count will be displayed for the report
- A calculation is made to determine how long ago the last incident was closed
- The process is initiated by clicking a button
- The same report that is displayed (without the record count) can be generated in MS Word
- All data comes from the **IncidentDownload** data source

#### What a Counter would see:

- There are two elementary processes: one to create the online report, and one to create the MS Word report
- Both the online and MS Word report has 4 standard pieces of repeatable information, 4 DETs
- The online report has a record count 1 DET, but the MS Word report does not
- A button click initiates each report, 1 DET
- There is one FTR, **IncidentDownload**
- The primary intent of both processes is to present information

#### Summation

- **EO with 1 FTR and 6 DETs giving it an Average complexity level and 5 Function Points**
- **EQ with 1 FTR and 5 DETs giving it an Low complexity level and 4 Function Points**

### 5.3.5. Business Requirement 05

#### BR 05 Background

BR 05 addresses the Individualized Questionably Aged Incident Reports. The user can review a list of incidents of questionable age, then set each record to be sent to any or all of the following: Test Lead, Creator, or Assigned To. The user can then kick off a process to generate the reports. One report will be created in MS Word for each person selected. (See Figure D4). Appendix G contains a sample of the individualized incident export.

#### BR 05 Analysis

The following fields are shown on the grid: Project ID, Project Name, Incident Number, Incident Description, Test Lead, Send to Test Lead, Creator, Send to Creator, Assigned To, and Send to Assigned To. There are two things happening on the "Create Reports" tab. First, the user is updating records in the database on the **IncidentDownload** table to designate who should know about the record. Second, the user can create these reports if and when they choose. The second process is associated, but not required, so it would be tested as a separate piece. The grid is not used for displaying an online report. The grid is used to display selection choices. It should be handled like a combo box or list box. The contents need to be tested (and counted), but that can be handled along with the other controls.

The traceability matrix for BR 05 would look like this:

Bus. Req.	Fun. Req.	Script
BR 05	FR 01	02.203.01 – Set Send To Flags
	FR 02	02.205.05 – Individual Incident Reports
	FR 03	

#### What a Tester would see:

- There are two pieces to test: updating the records in the **IncidentDownload** table, and creating the individualized reports from the same table
- The grid is used to select which records to change
- The checkboxes are used to select which fields to change (3 Send To fields)
- The process is initiated by clicking a button
- The reports have header information contained in table which are not updated by an application, so test queries will be constructed to verify that it is correct
- There are 8 pieces of repeatable information on the report
- The name of the person for the report is passed by the application and written to the title of the report file
- No calculations are performed for the report
- The report process is initiated by clicking a button

#### What a Counter would see:

- There are two elementary processes
- There is an EI to update records
- There is an EQ to present information
- There is 1 FTR for each process
- The report header is generated from tables that are neither ILFs nor EIFs, so they will not be counted

- The update process has one key field and 3 edited fields (the Send To fields), 4 DETs
- The report has 8 pieces of information plus the name for the file title, 9 DETs
- Each process is initiated by a button click for 1 DET each

**Summation**

- **EI with 1 FTR and 5 DETs giving it a Low complexity level and 3 Function Points**
- **EQ with 1 FTR and 10 DETs giving it a Low complexity level and 3 Function Points**

**5.3.6. Navigation and Control Population**

That completes the Business and Functional Requirements, but the test planning is not finished yet (neither is the function point counting). When testing an application, I usually have a section for navigation and control population. This information may be supplied in some design documents, technical requirements, or in many cases it might be implied. If it can be traced back to a requirement, test it against that requirement in the traceability matrix. The navigation section is to pick up things that would otherwise be lost. The purpose is to go back through the application to make sure all the controls were tested. This can be lumped in one script or spread through several, depending on the situation. The remaining control tests to be addressed are listed in the matrix.

The traceability matrix for BR 05 would look like this:

Bus. Req.	Fun. Req.	Script
Navigation	“Exit” Button	03.301.01 – General Application Navigation and Population
	“Apply Selection Filter” Button	
	“Select Specific Fields” Button	
	“Detailed/General” Radio Buttons	
	Population of “Select a person’s name” drop-down list	
	Population of “Project ID” drop-down list	
	Population of “Created By” drop-down list	
	Population of “Test Lead” drop-down list	
	Population of “Assigned To” drop-down list	
	Population of “Create Reports” grid	

#### **What a Tester would see:**

- Verify that the buttons, tabs, etc perform their intended function.
- The 4 drop-down list boxes and the “Create Reports” grid are dynamically loaded selection lists which are populated from the data contained in the local database on the **DownloadIncidents** table

#### **What a Counter would see:**

- Any control that is merely used for navigation will not be counted
- The “Select a person’s name” drop-down list is an EQ with 1 FTR, 1 DET for the name, and 1 DET for the button to initiate it
- The “Project ID” drop-down list is an EQ with 1 FTR, 1 DET for the name, and 1 DET for the button to initiate it
- The “Created By” drop-down list is an EQ with 1 FTR, 1 DET for the name, and 1 DET for the button to initiate it
- The “Test Lead” drop-down list is an EQ with 1 FTR, 1 DET for the name, and 1 DET for the button to initiate it
- The Assigned To” drop-down list is an EQ with 1 FTR, 1 DET for the name, and 1 DET for the button to initiate it
- The “Create Reports” grid is an EQ with 1 FTR and 11 DETs

#### **Summation**

- **EQ with 1 FTR and 2 DET giving it a Low complexity level and 3 Function Points**
- **EQ with 1 FTR and 2 DET giving it a Low complexity level and 3 Function Points**
- **EQ with 1 FTR and 2 DET giving it a Low complexity level and 3 Function Points**
- **EQ with 1 FTR and 2 DET giving it a Low complexity level and 3 Function Points**
- **EQ with 1 FTR and 2 DET giving it a Low complexity level and 3 Function Points**
- **EQ with 1 FTR and 11 DETs giving it a Low complexity level and 3 Function Points**

## **6.0. Time to Wrap Up the Project**

The project is finished and it is time to start the Function Point count. If the person testing the system had counting training, the meeting could start with the grid contained in Appendix H. This would greatly expedite the counting process.

As an added advantage, now the Function Point transactions are mapped to requirements to determine coverage. The transactions are also mapped to test scripts, so Function Points can be reported on as Tested, Not Tested, Passed, Failed, etc. If needed, scripts could even be tailored to better follow Function Point transactions.

**Appendix A – Script Numbering Breakdown**

**01. Class**  
**101. Procedure**

<b><u>Security</u></b>		<b><u>Business Function</u></b>		<b><u>Navigation</u></b>		<b><u>Data Validation</u></b>		<b><u>Interface</u></b>		<b><u>Configuration</u></b>	
<b>01.</b>		<b>02.</b>		<b>03.</b>		<b>04.</b>		<b>05.</b>		<b>06.</b>	
<b>101.</b>	Log on	<b>201.</b>	Add	<b>301.</b>	GUI	<b>401.</b>	Compatibility	<b>501.</b>	Report Templates	<b>601.</b>	Application Install
<b>102.</b>	Log off	<b>202.</b>	Delete	<b>302.</b>	Key Strokes	<b>402.</b>	File Format	<b>502.</b>	Import	<b>602.</b>	Compatibility
<b>103.</b>	Verify Profile	<b>203.</b>	Edit	<b>303.</b>	Title Bar	<b>403.</b>	Data	<b>503.</b>	Export	<b>603.</b>	Screen Resolution
<b>104.</b>	Registration	<b>204.</b>	Print	<b>304.</b>	Links	<b>404.</b>	Inputs	<b>504.</b>	Data Bases	<b>604.</b>	Logging
<b>105.</b>	Setup Security	<b>205.</b>	Report	<b>305.</b>	Screen	<b>405.</b>	Outputs	<b>505.</b>	Response Time		
		<b>206.</b>	Display			<b>406.</b>	Conversion				
		<b>207.</b>	Error Handling			<b>407.</b>	Video / Image				

## Appendix B – System Requirements

**BR 01** - The application should build a table in a local database (to facilitate off-line working ability) that contains test incident information.

**FR 01.01** - Must interface with the Lotus Notes Test Tracking Database to download the following pieces of information:

Date Created, Date Closed, Status, Created By, Assigned To, Severity, Severity Name, Priority, Priority Name, Project ID, Project Name, Incident Number, Incident Description

**FR 01.02** – Must interface with the Project Information Database to determine the Test Lead for each record.

**FR 01.03** – Must calculate the age of each incident, then interface with the Aged Reporting Support Database to determine what the maximum allowable age is for each records' Severity, then determine which (if any) rule it has violated and create a reason string for any violations. This information should be stored on each record to aid in reporting performance.

**FR 01.04** – Must create three fields to designate sending messages about the incident to either the creator, the assigned responsible, or the test lead.

**FR 01.05** – Must record the time the data was last updated, and display it to the user.

**FR 01.06** – Should default to display the Aged Incident Grid after data has been updated.

**BR 02** – The user should be able to pull up the rules for aged incidents to view them at any time during use of the application.

**BR 03** – The application should be able to display a report showing Questionably Aged Incidents to show reports that may need review.

**FR 03.01** – Only incident reports with aged violations will be shown

**FR 03.02** – The following fields should be shown on the report: Project ID, Project Name, Reason for Violation, Incident Number, Incident Description, Severity, Status, Age (in days), Test Lead, Creator, and Assigned To. The user should be able to select which of these fields are included in the report. A count of records should also be shown.

**FR 03.03** – The user should be able to filter the report two different ways.

**A** – Select a person's name to view a list of incidents where that person was the Test Lead, Creator, or Assigned To. The list of names should only contain choices that will return data.

**B** – Select any combination of the following: Project ID, Creator, Test Lead, or Assigned To. The lists should only contain choices that will return data.

**FR 03.04** – The user should be able to export the report to MS Word.

**BR 04** – The application should be able to display a report showing Questionably Aged Projects to show projects where records may need to be archived.

**FR 04.01** – Only projects that could be archived should be shown (all incident reports closed over 60 days).

**FR 04.02** – The following fields should be shown on the report: Project ID, Project Name, Test Lead, Days Since last Incident report was closed. A count of records should also be shown.

**FR 04.03** – The user should be able to export the report to MS Word.

**BR 05** – The application should be able to produce reports in MS Word (on the user's machine to facilitate off-line working ability) to be individually tailored to a person that will inform them of Questionably Aged Incidents.

**FR 05.01** – The user should be able to identify which incidents will be reported and to whom they will be reported (Test Lead, Creator, or Assigned To).

**FR 05.02** – The reports should inform the recipient what the purpose of the report is, what are the rules for Questionable Aged Incidents, why they are getting the report, and what they should do next.

**FR 05.03** – The report should contain the following data: Project ID, Project Name, Reason for Violation, Incident Number, Incident Description, Age (in days), Creator, Assigned To, and Test Lead. Group the data by Project ID and Project Name, then Reason for Violation. Sort the remainder of the data by Incident Number.



## Appendix C – Data Tables

Data Source: **Aged Reporting Support Database**

**Table: ActiveStatus**

	Field Name	Data Type
🔑	ID	AutoNumber
	ActiveStatusName	Text

**Table: InactiveStatus**

	Field Name	Data Type
🔑	ID	AutoNumber
	InactiveStatusName	Text

**Table: IncidentAgeLimits**

	Field Name	Data Type
🔑	ID	AutoNumber
	Severity	Text
	SeverityName	Text
	MaxAgeNew	Number
	MaxAgeActive	Number

**Table: ReportText**

	Field Name	Data Type
	ID	Text
	Message	Memo

Data Source: **Local Aged Database**

**Table: IncidentDownload**

	Field Name	Data Type
	DateCreated	Date/Time
	DateClosed	Date/Time
	Status	Text
	CreatedBy	Text
	AssignTo	Text
	Severity	Text
	SeverityName	Text
	Priority	Text
	PriorityName	Text
	ID	Text
	ProjectName	Memo
	IncidentNum	Text
	Description	Memo
	TestLead	Text
	Age	Number
	MaxAge	Number
	AgeViolation	Number
	Reason	Memo
	SendToCreatedBy	Text
	SendToAssignTo	Text
	SendToTestLead	Text

**Table: TimeStamps**

	Field Name	Data Type
	Event	Text
	Time	Date/Time

Data Source: **Project Information Database**

Table: **Projects**

	Field Name	Data Type
🔑	ProjNum	Text
🔑	ProjName	Text
	ProjTestLead	Text

Data Source: **Test Tracking Database**

Table: **IncidentReports**

	Field Name	Data Type
🔑	IncidentNum	Text
	DateCreated	Date/Time
	DateClosed	Date/Time
	Status	Text
	CreatedBy	Text
	AssignTo	Text
	Severity	Text
	SeverityName	Text
	Priority	Text
	PriorityName	Text
	ID	Text
	ProjectName	Memo
	Description	Memo

## Appendix D – GUI Interface

**Figure D1:** “List of Items” tab after data has been downloaded. The Questionably Aged Incident Report is displayed.

The Incident Data was last updated: 4/21/2004 4:01:54 PM

Get Data from Lotus Notes    Aging Rules    Quit

List of Items    Filter The Incidents    Create Reports

Reload Questionably Aged Incidents    Load Questionably Aged Projects    Export to Word    There are 83 records shown.

ID	ProjectName	Reason	IncidentNum	Description	SeverityName	Status	Age	TestLead	CreatedBy
S00005	Transportation	'High' se	7823.	Technical Fi	High	In Dev	30.	Rob G. Duchild	Susan A. Santel
S00005	Transportation	'High' se	7873.	Problems wit	High	In Dev	21.	Rob G. Duchild	Bruce A. Blunt
S00005	Transportation	'High' se	7723.	Technical Fi	High	In Tes	45.	Rob G. Duchild	Jessica B. Niehaus
S00005	Transportation	'High' se	7741.	Research: C	High	Reope	44.	Rob G. Duchild	Susan A. Santel
S00005	Transportation	'Low' se	7764.	Procedure: C	Low	New	42.	Rob G. Duchild	Susan A. Santel
S00005	Transportation	'Low' se	7810.	Fulfill - Order	Low	New	35.	Rob G. Duchild	Susan A. Santel
S00005	Transportation	'Medium	7765.	vice Crashed	Medium	In Dev	42.	Rob G. Duchild	Rick S. Guffey
S00005	Transportation	'Medium	7806.	Ship Verify S	Medium	In Dev	36.	Rob G. Duchild	Rick S. Guffey
S00005	Transportation	'Medium	7812.	Technical Fi	Medium	In Dev	35.	Rob G. Duchild	Susan A. Santel
S00005	Transportation	'Medium	7739.	Research: C	Medium	New	44.	Rob G. Duchild	Susan A. Santel
S00005	Transportation	'Medium	7798.	Freight order	Medium	New	36.	Rob G. Duchild	Rick S. Guffey
S00006	Vet Marketing	'High' se	7737.	Epsilon FTP	High	New	44.		Caroline E. Gunter
S00006	Vet Marketing	'High' se	7748.	Test Data fo	High	New	43.		Caroline E. Gunter
S00006	Vet Marketing	'High' se	7788.	Product Hier	High	New	37.		Caroline E. Gunter
S00007	Long Term PD	'Medium	7714.	Can NPPC s	Medium	New	48.		Sandra S. Eslinger
S00007	Long Term PD	'Medium	7715.	Common Ter	Medium	New	48.		Sandra S. Eslinger
S00007	Long Term PD	'Medium	7801.	Retailer attib	Medium	New	36.		Debbie A. Hendersc

**Figures D2A and D2B:** "Filter The Incidents" tab showing filter options A and B (respectively, from requirements) along with selections of which fields to display on the report.

The screenshot shows the 'Filter The Incidents' tab in the 'Obtain Questionable-Aged Incident Information' application. The interface includes the following elements:

- Header:** TESTING SERVICES GROUP logo.
- Status Bar:** The Incident Data was last updated: 4/21/2004 4:01:54 PM. Buttons: Get Data from Lotus Notes, Aging Rules, Quit.
- Navigation:** List of Items, Filter The Incidents (selected), Create Reports.
- Buttons:** Apply Selection Filter, Select Specific Fields, Update Questionably Aged Incident List.
- Make your filter selections:**
  - Radio buttons for  Detailed and  General.
  - Label: Select a person's name: [Dropdown Menu]
  - Dropdown menu options: Annamarie V. Erba, April G. Walgren, Audrey E. Wolff, Bruce A. Blunt, Carol D. Boyer, Carol M. Becker, Caroline E. Gunter, Charles G. Gilmore.
- Make your field selections:**
  - Checked checkboxes: ID, Reason, Description, Status, TestLead, AssignTo, ProjectName, IncidentNum, SeverityName, Age, CreatedBy.

The screenshot shows the 'Filter The Incidents' tab in the 'Obtain Questionable-Aged Incident Information' application with a different configuration:

- Header:** TESTING SERVICES GROUP logo.
- Status Bar:** The Incident Data was last updated: 4/21/2004 4:01:54 PM. Buttons: Get Data from Lotus Notes, Aging Rules, Quit.
- Navigation:** List of Items, Filter The Incidents (selected), Create Reports.
- Buttons:** Apply Selection Filter, Select Specific Fields, Update Questionably Aged Incident List.
- Make your filter selections:**
  - Radio buttons for  Detailed and  General.
  - Fields: Project ID, Created By, TSG Test Lead, Assigned To (each with a dropdown menu).
  - Dropdown menu for 'Assigned To' is open, showing options: Charles Gilmore, Chris E. Francis, Chris H. Kattentidt, Julie B. Pfister, Rob G. Duchild.
- Make your field selections:**
  - Checked checkboxes: ID, Reason, Description, Status, TestLead, AssignTo, ProjectName, IncidentNum, SeverityName, Age, CreatedBy.

**Figure D3:** "List of Items" tab after [Load Questionably Aged Projects] button has been clicked. The Questionably Aged Project Report is displayed.

The Incident Data was last updated: 4/21/2004 4:01:54 PM

Get Data from Lotus Notes    Aging Rules    Quit

List of Items | Filter The Incidents | Create Reports

Reload Questionably Aged Incidents    Load Questionably Aged Projects    Export to Word    There are 11 records shown.

ID	ProjName	TestLead	DaysAgo
S00002	Consolidate Risk Exposure - Phase I (PROJECT INCIDENTS)		212.
S10815	WMS Windows Migration	Carol D. Boyer	217.
S10835	Testing for CS Support	Carol D. Boyer	62.
S11040	One Sales Reporting - Daily Demand Front-End	Chris E. Francis	303.
S11041	One Sales Reporting - Daily Demand Back-End	Chris E. Francis	358.

**Figure D4:** "Create Reports" tab while updating whom to send reports.

The Incident Data was last updated: 4/21/2004 4:01:54 PM

Get Data from Lotus Notes    Aging Rules    Quit

List of Items    Filter The Incidents    **Create Reports**

Create Reports    Update

Send To 'TestLead'  
 Send To 'CreatedBy'  
 Send To 'AssignTo'

Project	IncidentNum	Description	TestLead	SendToTestLead	CreatedBy	SendToCreatedBy	AssignTo	SendToAssignTo
Transpc	7723.	Technical Fi	Rob G. Duc	no	Jessica B. N	no	Lori L. Bulle	no
Transpc	7739.	Research: C	Rob G. Duc	no	Susan A. Se	no	Jerry Shield:	no
Transpc	7741.	Research: C	Rob G. Duc	no	Susan A. Se	no	Jane L. Gre	no
Transpc	7764.	Procedure: (	Rob G. Duc	no	Susan A. Se	no	Jessica B. M	no
Transpc	7765.	Ice Crashed	Rob G. Duc	no	Rick S. Guff	no	Matthew A.	no
Transpc	7798.	Freight order	Rob G. Duc	no	Rick S. Guff	no	Jerry Shield:	no
Transpc	7806.	Ship Verify S	Rob G. Duc	no	Rick S. Guff	no	Matthew A.	no
Transpc	7810.	Fulfill - Order	Rob G. Duc	no	Susan A. Se	no	Jerry Shield:	no
Transpc	7812.	Technical Fi	Rob G. Duc	no	Susan A. Se	no	Vickie L. Re	no
Transpc	7823.	Technical Fi	Rob G. Duc	no	Susan A. Se	no	Jessica B. M	no
Transpc	7873.	Problems wil	Rob G. Duc	no	Bruce A. Blu	no	Caroline E. I	no
Vet Mar	7737.	Epsilon FTP		no	Caroline E. C	no	Rita G. Kircl	no
Vet Mar	7748.	Test Data fc		no	Caroline E. C	no	Caroline E. I	no
Vet Mar	7788.	Product Hier		no	Caroline E. C	no	Caroline E. I	no
Long Te	7714.	Can NPPC s		no	Sandra S. E	no		no
Long Te	7715.	Common Te		no	Sandra S. E	no		no

## Appendix E – Sample report for Questionably Aged Incident MS Word Export

ID: S00005  
ProjectName: Transportation Implementation Issues  
Reason: 'High' severity, 'In Development / Research' status incident over 20 days old.  
IncidentNum: 7823.  
Description: Technical Fix: Need carrier response notification for EDI carriers  
SeverityName: High  
Status: In Development / Research  
Age: 30.  
TestLead: Rob G. Duchild  
CreatedBy: Susan A. Santel  
AssignTo: Jessica B. Niehaus

ID: S00005  
ProjectName: Transportation Implementation Issues  
Reason: 'High' severity, 'In Development / Research' status incident over 20 days old.  
IncidentNum: 7873.  
Description: Problems with ALOCRATE AND VOLAPGL 'touching' Every Row in COHFPTBL Table  
SeverityName: High  
Status: In Development / Research  
Age: 21.  
TestLead: Rob G. Duchild  
CreatedBy: Bruce A. Blunt  
AssignTo: Caroline E. Gunter

## Appendix F – Sample report for Questionably Aged Project MS Word Export

ID: S00002  
ProjName: Consolidate Risk Exposure - Phase I (PROJECT INCIDENTS)  
TestLead:  
DaysAgo: 283.

ID: S10815  
ProjName: WMS Windows Migration  
TestLead: Carol D. Boyer  
DaysAgo: 288.

ID: S10835  
ProjName: Testing for CS Support  
TestLead: Carol D. Boyer  
DaysAgo: 133.

ID: S11040  
ProjName: One Sales Reporting - Daily Demand Front-End  
TestLead: Chris E. Francis  
DaysAgo: 374.



## Questionably Aged Incident Report

This report was created for the purpose of automating the review of incidents in the TSG Test Tracking Database, and calling out those whose age has become questionable. What is the criterion for an incident to be of questionable age?

**'Urgent' severity incidents over 5 days old with a 'New' status, or over 15 days old with any active status.**  
**'High' severity incidents over 10 days old with a 'New' status, or over 20 days old with any active status.**  
**'Medium' severity incidents over 15 days old with a 'New' status, or over 30 days old with any active status.**  
**'Low' severity incidents over 20 days old with a 'New' status, or over 45 days old with any active status.**  
**'None' severity incidents over 25 days old with a 'New' status, or over 60 days old with any active status.**

You are being sent this message, because you are either:

- 1) The TSG Test Lead for the project where the incident was created
- 2) The Creator (or author) of the incident
- 3) The person to whom the incident was assigned

Please review the incident(s) listed in this report, and update their status or severity if appropriate. If you have any questions, please feel free to contact a member of the TSG, or call Chris Francis (x3502).

### **S00005- Transportation Implementation Issues**

**Reason: 'Medium' severity, 'In Development / Research' status incident over 30 days old.**

7765- 03/10/2004 - Line Item Update Import Service Crashed

Age: 42

Created By: Rick S. Guffey

Assigned To: Matthew A. Mertens

Test Lead: Rob G. Duchild

7806- Ship Verify Service - Freight order not found

Age: 36

Created By: Rick S. Guffey

Assigned To: Matthew A. Mertens

Test Lead: Rob G. Duchild

**Appendix H – Final Traceability Matrix**

Information Needed for Testing			Shared Needs for Function Point Counting		Counting Specific	
Bus. Req.	Fun. Req.	Script	EI/EO/EQ	FTR / DET	Complexity	Function Points
BR 01	FR 01	04.403.01 – Import Incident Data	EI	4 / 16	High	6
	FR 02					
	FR 03					
	FR 04					
	FR 05					
	FR 06	(See BR 03)	---	--- / ---	---	---
BR 02	N/A	04.403.02 – Display Aging Rules	---	--- / ---	---	---
BR 03	FR 01	02.205.01 – Aged Incident (Unfiltered)	EO	1 / 24	Average	5
	FR 02	02.205.02 – Aged Incident (General Filter)				
	FR 03	02.205.03 – Aged Incident (Detailed Filter)	EQ	1 / 23	Average	4
	FR 04	05.503.01 – Export Incident Report to MS Word				
BR 04	FR 01	02.205.04 – Aged Projects	EO	1 / 6	Average	5
	FR 02	05.503.02 – Export Project Report to MS Word	EQ	1 / 5	Low	4
	FR 03					
BR 05	FR 01	02.203.01 – Set Send To Flags	EI	1 / 5	Low	3
	FR 02	02.205.05 – Individual Reports	EQ	1 / 10	Low	3
	FR 03					
Navigation	Navigation	03.301.01 – General Application Navigation and Population	---	--- / ---	---	---
	Population of “Select a person’s name” drop-down list		EQ	1 / 2	Low	3
	Population of “Project ID” drop-down list		EQ	1 / 2	Low	3
	Population of “Created By” drop-down list		EQ	1 / 2	Low	3
	Population of “Test Lead” drop-down list		EQ	1 / 2	Low	3
	Population of “Assigned To” drop-down list		EQ	1 / 2	Low	3
	Population of “Create Reports” grid		EQ	1 / 11	Low	3

Information Needed for Testing		Shared Needs for Function Point Counting		Counting Specific	
Data Sources	Data Validation Testing	ILF/EIF	RETs / DETs	Complexity	Function Points
ActiveStatus	No	---	--- / ---	---	---
InactiveStatus	No	---	--- / ---	---	---
IncidentAgeLimits	No	---	--- / ---	---	---
ReportText	No	---	--- / ---	---	---
Projects	No	EIF	1 / 3	Low	5
IncidentReports	No	EIF	1 / 13	Low	5
IncidentDownload	Yes	ILF	1 / 21	Low	7
TimeStamps	Yes	ILF	1 / 2	Low	7

**Unadjusted Function Point Count = 72**