# *Black Box Testing – Puerto Rico Tax Form 482 using "Planilla Plus+" Software*

*Emmanuel Claudio Valentín*
*Master of Engineering in Computer Engineering*
*Alfredo Cruz, PhD*
*Electrical and Computer Engineering and Computer Science Department*
*Polytechnic University of Puerto Rico*

**Abstract** — *Software testing is a very important development process. Every software design must have a development testing phase. Although, not all developers apply testing to their software. The phase of testing a software is required to point out the defects and errors that were made during the development phases. In this project, we use Black Box testing methodology to test "Planilla Plus+" software, using Puerto Rico Tax Form 482. In Puerto Rico exist many tax forms software to help people to fill their tax forms. Not everybody knows how to fill a tax form from scratch. The idea of this software is to help people that don't have any knowledge how to fill out their tax forms without any professional help since now the electronic tax form submission is required by law in Puerto Rico starting 2016. This project answer these questions applying Black Box Testing Methodology.*

***Key Terms*** — *Black Box Testing, Software Testing, Test Cases, Testing Techniques.*

## INTRODUCTION

Software testing is one of the most used techniques for verifying and validation of a software. Is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating and verifying that a software program or application or product meets the business and technical requirements that guided its design and development [1].

Verification and validation are the practices that compose software testing. Verification makes sure that the software or product is designed to deliver all functionality to the customer. Is done at the starting of the development process. Validation determine if the systems complies with the requirement and performs functions for which it is intended and meets the organizations goal and user needs. Is normally done at the end of development process and takes places after verification.

They are many different techniques that exist to test a software. In our case, we use Black Box testing. Black Box testing is a technique of testing without having any knowledge of the interior workings of the application. The tester is oblivious to the system architecture and does not have access to the source code. Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon [2].

Black Box testing have some characteristics worth to mention. First, the testing is made by persons that use the system, not the developers of the application and have tools to facilitate testing procedures. It is a software test design technique that involves dividing input values into valid and invalid partitions and selecting representative values from each partition as test data. Involves determination of boundaries for input values and selecting values that are at the boundaries and just inside outside of the boundaries as test data and involves identifying the cases (input conditions) and effects (output conditions), producing a Cause-Effect Graph, and generating test cases accordingly.

This project used Black Box testing methodology to test "Planilla Plus+" Software 2015 using Tax Form "Planilla Unica 482" from Puerto Rico [3].

## PROBLEM STATEMENT

There's always been a problem when we fill out tax forms using software. Some people leave parts that need to be completed, some has problems to understand how the software works, others don't like how the software is organize or seems to be

confusing how is presented. Now, can a person without any knowledge can fill out correctly their tax form using a software? These are the type of questions and problems we what to investigate. There are many reasons for software be addressed the wrong way, but there are others software that developers take the time to analyze and create an easy and complete software that fulfill its mission of fill out a tax form correctly using testing methodologies.

This methodology is focused of the user interface area that is the most critical part to design in a software. If the user interface is not a good design, the software will be difficult to understand and use.

## BLACK BOX TESTING - METHODOLOGY

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. For every testing phase, we will present in details the steps applied. This testing method is applied to "Planilla Plus+" software to PR tax form "Planilla Unica 482" 2015 of Puerto Rico. To develop a functional Black Box testing we will follow these steps:

- Data Flow Testing
- Control Flow Testing
- Transaction Flow Testing
- Domain Testing
- Syntax Testing
- Finite-State Testing
- Loop Testing

### Data Flow Testing

Data Flow testing uses the control graph flow graph to explore the unreasonable things that can happen to data. Consideration of data flow anomalies leads to test path selection strategies that fill the gaps between complete path testing and branch and statement testing [4].

The name of data flow testing is given to a family of test strategies based on selecting paths through the program control flow to explore sequences of events related to the status of data objects. The data flow graph is a graph consisting of nodes and directed links.

We apply dataflow testing tool to detect improper use of data values due coding errors. We will focus on the points at which variables receives values and the points at which these values are used or referenced.

Test design and execution is the same as for control-flow models, with some minor differences that result from having data-flow graphs rather than control-flow graphs.

1. Always verify the specification.
2. Identify input variables, especially constants.
3. List the functions, starting with those that depend only on input variables.
4. Examine intermediate functions and see if the sequencing is essential or merely convenient.
5. Conversely, you may be able to simplify the model by adding an intermediate node for a complicated calculation.
6. Verify the model.

### Control Flow Testing

Behavioral control-flow testing was introduced as the fundamental model of black-box testing. The control-flow graph is the basic model for the test design [4]. Control-flow behavioral testing is a fundamental testing technique that is applicable to majority of software programs and is quite effective for them. It is generally applicable for comparatively smaller programs or even for smaller segments of bigger programs.

Test design begins by creating a behavioral control-flow graph model from requirements documents such as specifications. The list notation is generally more convenient than graphical graphs, but small graphs are an aid to model design. Control flow testing is applied to build and present graph that represents the path to take in Planilla Plus+ software. Test design technique and execution consists of the following steps:

1. Examine the requirements and analyze them for operationally satisfactory completeness and self-consistency.
2. Rewrite specification as a sequence.
3. Build the model.
4. Select test paths.
5. Predict, record expected outcome for each test.
6. Run the tests.
7. Confirm the paths

### Transaction Flow Testing

The transaction-flow graph is used in system testing of on-line applications and batch-processing software. It has both control-flow and data-flow attributes [4]. All software applications are built to perform a set of transactions to move data from inside to outside the application boundary or outside to inside the application boundary with or without mathematical or logical operations. Thus, software system is a collection of transactions that represent dynamic behavior of the system by moving the data around. Test design technique and execution consists of the following steps:

1. Verify the specification.
2. Identify and name all the transactions.
3. Define a hierarchy of transaction types.
4. Define the transaction state.
5. Identify how every transaction is: born, dies, merges, absorptions, splits, births, and so on.
6. Identify all queues.
7. Identify processing components.
8. Select your test "paths" they have aspects of both paths and slices.

### Domain Testing

Domain testing is used to test software or portions of software dominated by numerical processing. It replaces the common heuristic method of testing extreme values and limit values of inputs. It is a formal and automatable technique to replace the common practiced of testing. Domain Testing on formally defining process domains assets of boundary into a space [4]. Domain Testing is used to show graph were numerical process apply and build a model based in the test performed for the systems screens that apply.

### Syntax Testing

Syntax testing is a powerful technique for testing command-driven software and similar applications. It is easy to do and is supported by commercial tools [4]. It's a lexical analyzer and parser of the command driven software. The techniques to perform syntax testing:

1. Gather specification for commands to be tested.
2. Find and group common parts to apply for many commands.
3. Find and define keywords.
4. Create specification for command fields.
5. Group commands.

### Finite State Testing

Originally motivated by hardware logic testing, the finite-state machine model is an excellent model for testing menu-driven applications. It is also important because of its widespread use in object-oriented design [4]. A finite state machine is a model to describe the dynamic behaviors of an object over time. A finite state machine is a localized view of an object. Each object is treated as an isolated entity that communicates with the rest of the world by detecting events and responding to them. Finite state verification techniques are intermediate in power and cost between construction of simple control and data flow models, on the one hand, and reasoning with the full strength of symbolic execution and theorem proving on the other. The technique to perform Finite State Testing include:

1. Identify inputs.
2. Define input encoding.
3. Identify States.
4. Define the state encoding.
5. Identify output events.
6. Define the output encoding.
7. Build the state table and output table and clean them up.
8. Design your test.

9. Run the test.
10. For every input confirm: transition, output.

### Loop Testing

Loop testing is a heuristic technique that should be used in conjunction with many other testing methods because experience shows that bugs often accompany loops. The techniques discussed in this apply when there are loops in a graph, such as: control-flow graph, transaction-flow graph, or syntax graphs [4]. The types of loops are nested loops, unstructured loops, nondeterministic loops, and deterministic loops.

## BLACK BOX TESTING – RESULTS

These are the results found after Black Box Testing was applied to "Planilla Plus+" Software.

### Data Flow Testing

Below are the results of Data Flow testing applied to Planilla Plus+ software for the tax form 482 from PR. Every result is shown in ascendant order of the application sequence. The test is divided in sections by the name representing data groups.

### Incomes

*W-2PR Form Incomes* – Equation (1) represent the calculations of wages, incomes, tips, etc in W-2PR Form. All the reported incomes must be sum to fill out correctly the Tax Form 482 Section 3 of Encasillado 1 in Tax Form Planilla Unica. The total is composed of the sum of four inputs: Wages, Commissions, Allowances, and Tips.

*Total of W-2PR Incomes* = Wages + Commissions + Allowances + Tips (1)

*Others Incomes* - The application has inputs values across different forms. He saves the values in each form and in the final step made the calculations of the total of all forms incomes. The total of incomes is made by the sum of many forms inputs show below in Equation (3). The total of all Incomes Forms is located in Section 3 Block 1 of Planilla Unica 482 Tax Form of PR.

*Total of Incomes* = W-2PR Form Incomes + IE-I-8 Form Incomes + IE-II-1 Form Incomes + IE-II-4 Form Incomes + IE-II-9 Form Incomes + IE-II-20 Form Incomes + 480.6 EC Form Incomes + 480.6F Form Incomes + 480.7 Form Incomes + 482-2R Form Incomes + 482-4 Form Incomes + IE-I-7 Form Incomes + Anejo H Form + IE-II-13 Form Incomes + N Form Incomes + IE-I-6 Form Incomes + IE-II-7 Form Incomes + D1 Form Incomes + 1034.04 Form Incomes + Business Incomes + Interest Incomes + Dividends Incomes + Distributions Incomes + Capital Lost and Gains Incomes + Any Others Incomes (2)

### Deductions

The deductions are calculated by authorized forms of investment or payments by law. The application follows the same structure has before. The application has a section with a list of all form approved. All these forms represent the different authorized way to deduct incomes or payments in the Tax Form 482 Planilla Unica of Puerto Rico. Equation (3) show the data flow of the total of deductions in Tax Form 482 found in Anejo 2 Section 8 of Planilla Unica.

*Total of Deductions* = Mortgage Interest 480.7A Form + Principal Mortgage Interest 1098 Form + 30% Mortgage Exclusion A-I-1 or CO-5-A Form + Loan Points A-I-1 o CO-5-A Form + Mortgage Lost A-I-2 o CO-5-B Form + Loss of Furniture A-I-5 o CO-5-E Form + Dependents A1 and A-II o CO-15 Form + Medical Expenses A-III Form + Student Loan Interest A-I-10 o CO-6-E Form + Retirement IRA 480.7 Form + Government Retirement A-I-6 Form + Donatives A-III Form + Deposits to FCP-PR Form (3)

### Credits

The credits in tax forms are very simple. They take a credit and calculate a percent or formula depending the type of credit and make a total sum of all credits types reported in the year. This credits are used to sum the negative or positive of the returns or payments you have in the tax form calculations. In PR Tax Form 482 the credits can be found in Anejo B2 Individuo. Equation (4) represent the data flow of the total of credits for PR Tax Form Planilla Unica 482.

*Total of Credits* = Credit of American Opportunity (B2) + Credit of Increment Agro Products (Sección 1051.07) (Anejo B1 Parte I) + Credit for Products (4)

Made in Puerto Rico (Sección 1051.09) (Anejo B1 Parte III) + Credit for Products Made in Puerto Rico Process (B1 Parte IV, 6) + Contributions Pay to EEUU (C) + Credit Recovery by Investment (B Parte I) + Créditos por Fondos de Inversión (Q Parte 1) + Arrastre de pérdidas no utilizadas en años anteriores en Pérdidas en la Venta (Q Parte 4 Linea 2) + Total de pérdidas incurridas en cada uno de los años anteriores (Q Parte 4 Linea 4) + Crédito por inversión reclamado durante el año contributivo relacionado con la inversión sujeta a pérdida (Q Parte 4 Linea 10) + Crédito por Inversión en Infraestructura de Vivienda (B Parte II, 2) + Crédito por Inversión en la Construcción de Vivienda para Alquiler a Familias de Ingresos Bajos (B Parte II, 3) + Crédito por Inversión en Construcción en Centros Urbanos (B Parte II, 4) + Crédito para Comerciantes Afectados por la Revitalización de los Cascos Urbanos (B Parte II, 5) + Crédito por Constitución de Servidumbre de Conservación Elegible o Donación de Terreno Elegible (B Parte II, 7) + Crédito por Ley 135 de 1997 (B Parte II, 14) + Crédito por Inversión en Desarrollo Industria Fílmica (B Parte II, 15) + Crédito para Inversionistas que Adquieran un Negocio Exento que esté por Cerrar Operaciones en P.R. (B Parte II, 17) + Crédito por Inversión Ley 73 de 2008 (B Parte II, 19) + Crédito por Inversión Ley 83 de 2010 (B Parte II, 20) + Crédito por Pagos de Membresía por Miembros de Corporación Especial Propiedad de Trabajadores (B Parte II, 16) + Crédito por donativos al Patronato (B Parte II, 18) + Créditos Arrastrados de Años Anteriores Sujetos a Moratoria (B Parte II, 9) + Créditos Arrastrados de Años Anteriores No Sujetos a Moratoria (B Parte II, 22) + Compra Créditos (B Parte IV)

## Control Flow Testing

Below are the results of Control Flow testing applied to Planilla Plus+ software for the tax form 482 from PR. Every result is shown in ascendant order of the application sequence. The test is divided in six parts by their names representing data groups.

## Part I – Personal Information

The Part I covers all the personal information of the taxpayer in the year 2015.

### *Civil Personal State Form Model*

The Table 1 is a Build Model of Civil Personal State Form of PR Tax Form Planilla Unica 482. Represent the modeled version of the Civil Personal State Form. The model in Table 1 is a detail version of the requirements that are needed.

**Table 1**
**Model of Civil Personal State Form**

| Start Node | End Node | Description |
|---|---|---|
| 1 | 2 | Choose “Casado que rinde planilla …” |
| | 3 | Choose “Casado que rinde separado” |
| | 4 | Choose “Individual no casado” |
| | 5 | Choose “Individual casado con capitu…” |
| | 6 | Choose “Individual casado que no ...” |
| 7 | | Continue with model |

### *Taxpayer Personal Information Form Model*

The Table 2 is a Build Model of Taxpayer Personal Information Form of PR Tax Form Planilla Unica 482. Represent the modeled version of the Taxpayer Personal Information Form. The model in Table 2 is a detail version of the requirements that are needed.

**Table 2**
**Model of Taxpayer Personal Information Form**

| Start Node | End Node | Description |
|---|---|---|
| 1 | 2 | Input of “Nombre” |
| 2 | 3 | Input of “Initial” |
| 3 | 4 | Input of “Apellido Paterno” |
| 4 | 5 | Input of “Apellido Materno” |
| 5 | 6 | Input of “Sexo” |
| 6 | 7 | Input of “Seguro Social” |
| 7 | 8 | Input of “Fecha de Nacimiento” |
| 8 | 9 | Input of “Ocupacion” |
| 9 | 10 | Input of “Contrato de Gobierno” |
| 10 | 11 | Input of “Veterano de EU” |
| 11 | 12 | Input of “¿Es reclamado como depen…” |
| 12 | 13 | Input of “Ciudadano de Estados Unidos” |
| | 16 | Input of “Si el contribuyente falleció …” |
| | 18 | Input of “Si su cónyuge fallecio du…“ |
| | 20 | Input of “Seguro Social del cónyuge …” |
| 13 | | Input of “¿Militar activo en zona de …” |
| 13 | 14 | Input of “Si fué militar activo en zon…” |
| 14 | 15 | Input of “Fecha de cese” |
| 15 | | Continue with model |
| 16 | 17 | Input of “Fecha de defucion” |
| 17 | | Continue with model |
| 18 | 19 | Input of “¿Es usted el cónyuge supers…” |
| 19 | | Continue with model |
| 20 | 21 | Input of “Fecha de defunción del cón…” |
| 21 | 22 | Input of “Solicito Prorroga” |
| 22 | | Continue with model |

### *Postal Address Form Model*

The Table 3 is a Build Model of Postal Address Form of PR Tax Form Planilla Unica 482. Represent the modeled version of Postal Address

Form. The model in Table 3 is a detail version of the requirements that are needed.

**Table 3**
**Model of Postal Address Form**

| Start Node | End Node | Description |
|---|---|---|
| 1 | 2 | Input of "Formato" |
| 2 | 3 | Input of "Linea 1" |
| 3 | 4 | Input of "Linea 2" |
| 4 | 5 | Input of "Ciudad" |
| 5 | 6 | Input of "Estado" |
| 6 | 7 | Input of "Codigo Postal" |
| 7 | 8 | Input of "Es un cambio" |
| 8 | | Input of "Fisica igual a Postal" |

## Paths

For better understanding P# means Possible Path followed by the number sequentially.

### *Civil Personal State Form*

These are all the possible paths we can take from the Model of Civil Personal State Form in Table 1.

P1: 1,2,7 P2: 1,3,7 P3: 1,4,7

P4: 1,5,7 P5: 1,6,7

The start and end paths are the same the main difference is the middle option that that represent the tax payer civil status. The civil status can only be one of the options.

### *Taxpayer Personal Information Form*

These are all the possible paths that can take from the Model of Taxpayer Personal Information Form in Table 2.

After analyzing the model in Table 2, nine possible solutions were found. Here we can find many inputs that decide whatever you may or not skip some inputs like option 13.

### *Postal Address Form*

These are all the possible paths we can take from the Model of Postal Address Form in Table 3. Only one possible path exists in this Form.

P1: 1, 2, 3, 4, 5, 6, 7, 8

## Part II – *Informatives*

Part II covers all the informative documentation forms of the taxpayer. Here the taxpayer must select and add all information from the different documentations forms about their incomes and expenses that apply in Puerto Rico law.

Graph - The W-2PR Form reports an employee annual wages and the amount of taxes withheld from paycheck in Puerto Rico. The W-2PR Form can be as many job incomes the taxpayer have.

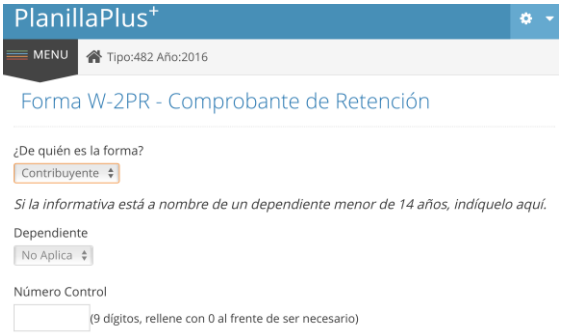


**Figure 1**
**Print Screen of W-2PR Form**

The following graph represents the information inputs of each node in Figure 2 of W-2PR Form. Detail information about each node can be found in Table 4, Model of W-2PR Form.

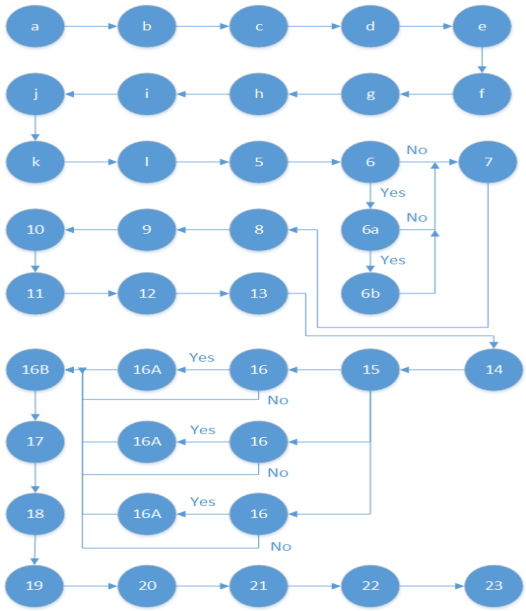


**Figure 2**
**Graph of W-2PR Form**

Paths - Based on the model of W-2PR in Table 4, these are all the possible paths.

P1: a, b, c, d, e, f, g, h, i, j, k, l, 5, 6(yes), 6a, 6b, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16(yes), 16A, 16B, 17, 18, 19, 20, 21, 22, 23

P2: a, b, c, d, e, f, g, h, i, j, k, l, 5, 6(no), 7, 8, 9, 10, 11, 12, 13, 14, 15, 16(yes), 16A, 16B, 17, 18, 19, 20, 21, 22, 23

P3: a, b, c, d, e, f, g, h, i, j, k, l, 5, 6(yes), 6a, 6b, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16(no), 17, 18, 19, 20, 21, 22, 23

P4: a, b, c, d, e, f, g, h, i, j, k, l, 5, 6(no), 7, 8, 9, 10, 11, 12, 13, 14, 15, 16(no), 17, 18, 19, 20, 21, 22, 23

W-2PR Form have only four possible paths. Depending of the choices made in node 6 and 16 will determine the path to take.

## Part III – *Incomes*

Part III is all about tax payer incomes. Here the taxpayer adds all the inputs for incomes in the taxable year. The application simplifies grouping incomes types to report. These are salaries, family, pensions, unemployment, rent, wins, business, interests, distributions, lost and earnings and any other incomes.

### *Salaries Group Inputs Options Model*

The Table 5 is a Build Model of Salaries Group Inputs Options of Incomes Main Form. Represent the modeled version of Salaries Group Inputs Options.

**Table 5**
**Model of Salaries Group Inputs Options**

| Start Node | End Node | Description |
|---|---|---|
| 1 | 3 | Link to “Formas W-2PR: Comprobante …” |
| | 2 | W-2PR Form |
| 3 | 5 | Link to “Formas W-2: Comprobante de …” |
| | 4 | W-2 Form |
| 5 | 7 | Link to “Cantidades pagadas por un pat…” |
| | 6 | IE-I-8 Form |
| 7 | 9 | Link to “Beneficios marginales pagados…” |
| | 8 | IE-II-1 Form |
| 9 | 11 | Link to “Gastos de sacerdotes o minist…” |
| | 12 | IE-II-9 Form |
| 11 | 13 | Link to “Aguinaldo de Navidad, Bono d…” |
| | 14 | IE-II-20 |

### *Family Incomes Groups Model*

The Table 6 is a Build Model of Family Incomes Group of Incomes Main Form. Represent the modeled version of Family Incomes Group.

**Table 6**
**Model of Family Incomes Group**

| Start Node | End Node | Description |
|---|---|---|
| 1 | 3 | Link to “Pensión Recibida por Divorc …” |
| | 2 | 482-2R Form |
| 3 | 5 | Link to “Pensión Pagada por Divorcio…” |
| | 4 | 482-4 Form |
| 5 | 7 | Link to “Pensión alimenticia a menores” |
| | 6 | IE-I-7 Form |

### *Pensions, Unemployment, Rent, Wins, and Mortgages Incomes Groups Model*

The Table 7 is a Build Model of Pensions, Unemployment, Rent, Wins, and Mortgages Incomes Groups of Incomes Main Form. Represent the modeled version of Pensions, Unemployment, Rent, Wins, and Mortgages Incomes Groups.

**Table 7**
**Model of Pensions, Unemployment, Rent, Wins, and Mortgages Incomes Groups**

| Start Node | End Node | Description |
|---|---|---|
| 1 | 2 | Link to “Pensiones y Anualidades” |
| | 1a | Anejo H Form |
| 2 | 3 | Link to “Desempleo” |
| | 2a | IE-II-13 Form |
| 3 | 4 | Link to “Renta” |
| | 3a | N Form |
| 4 | 5 | Link to “Premios” |
| | 4a | IE-I-6 Form |
| | 4b | IE-II-7 Form |
| 5 | | Link to “Residencia” |
| | 5a | D1 Form |
| | 5b | D3 Form |

### *Salaries Group Inputs Options Paths*

After analyzing the model in Table 5 we found 18 possible solutions. The model contains 14 nodes that include many decisions that affects each path.

### *Family Incomes Group Paths*

After analyzing the model in Table 6 we found 7 possible solutions. The model contains 7 nodes. This group is very simple and easy to understand.

### *Pensions, Unemployment, Rent, Wins, and Mortgages Incomes Groups Paths*

After analyzing the model in Table 7 it is found 12 possible permutation for solutions equals to 479001600. The model contains 12 nodes. This group is very complicated and difficult to understand.

### Part IV – *Deductions*

Part IV is all about tax payer deductions. Here the taxpayer adds all the inputs for deductions in the taxable years. The application simplifies using groups depending of the deductions like previous parts. These are mortgage, family, health, education, retirement, donations, and other deductions. are going to explain each group at a time for easier understanding.

*Paths* - After analyzing all inputs and creating a model for deduction inputs, we found 20! possible solutions equals to 2.432902e+18. These groups have many nodes. Again, another complicated way to present possible ways and inputs to the user.

### Part V – *Credits*

Part V is all about tax payer credits. Here the taxpayer adds all the inputs for credits in the taxable year. The application simplifies using groups depending of the credit type, same format used in previous parts. These are education, agriculture, contributions, rebuy, investments, other credits, and buy of credits.

*Paths* - After analyzing all inputs and creating a model for credits inputs, we found 32! possible solutions equals to 2.6313084e+35. These groups have many nodes. Again, another complicated way to present possible ways and inputs to the user.

### Part VI – *Payments*

In Part V the taxpayer adds all the inputs for payments to receive or previously made in the taxable year. The application simplifies using groups depending of type of payments to report. These groups are Reimbursement, Tax Form, Retentions, and Adjustments of Payments. The software in some cases refer to a form previously explained in Part II Informatives. In those sections, we did not create control flow since they are created already in Part II Informatives.

*Paths* - After analyzing all inputs and creating a model for payments inputs, we found 16! possible solutions equals to 2.092279e+13. The model contains 16 nodes that can be permutated in many ways. Very complicated model to a simple section to enter only payments information inputs.

### Transaction Flow Testing

Transaction is a unit of work seen from a system user's point of view. A transaction consists of a sequence of operations, some of which are performed by a system, persons or devices that are outside of the system. To represent the transaction flow testing of Planilla Plus+ software diagrams are used. Diagrams makes easier to understand the transactions made in the application. To create the diagrams, the specification and identify all the transactions of the application life cycle are verified. States and the flow of transactions are defined. Figure 3 show the Transaction Flow of the complete walkthrough of the application life cycle. The letters A through G represents subs-transaction explained below in Figure 4 and 5. These transactions represent in Figure 4 and 5 are different transactions made depending the section of the application.

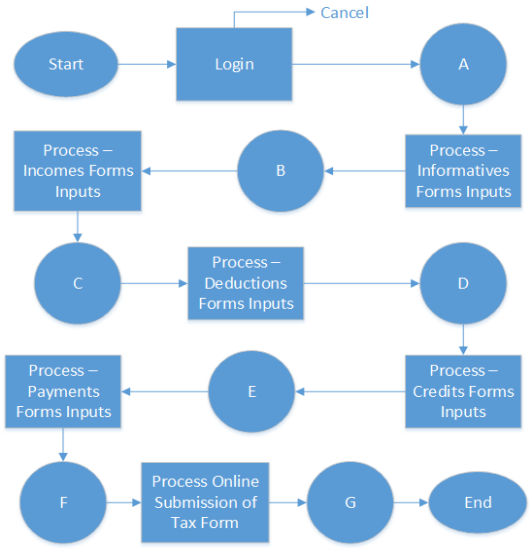


**Figure 3**
**Transaction Flow Diagram of the Application Life Cycle**

In Figure 4 we can observe that the application having different sections using different forms and inputs it maintains the same behavior of transactions. The application is reusing the transactions in each different section. Here the application receives the inputs and type of form and save it for later processing.

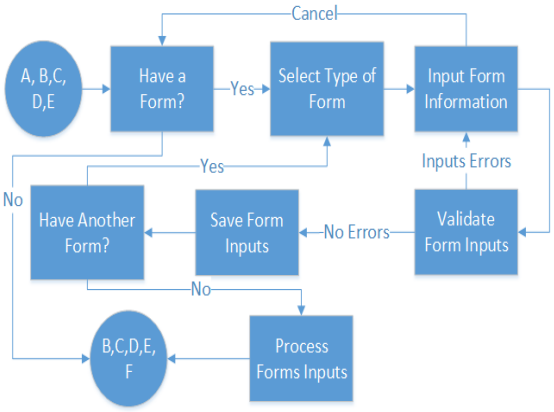


**Figure 4**
**Transaction Flow Diagram of Forms**

In Figure 5 we have a validation and submission of the transaction. In this section the application validate all the forms inputs and submitted the tax form if the inputs of all section forms (Figure 4) has no error. If the application finds an error, it will tell you where the error is located to go and fix it. Finally, the application let you make a copy of the tax form previously submitted for records and end the all transactions.

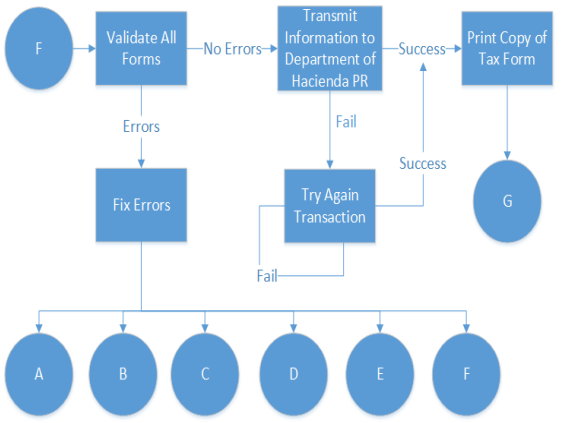


**Figure 5**
**Transaction Flow Diagram for Validation and Processing Forms**

## Domain Testing

Domain testing is a family of test strategies based on selecting paths through the program's control flow to explore sequences of events related to the status of variables or data objects. Tests are created by selecting slices from the output nodes to all the input nodes of the slice [4]. Unfortunately, the Planilla Plus+ application don't have or show any status of what happens with the processed inputs. How the data flow is being progressed through the application with out and output cannot be tested. For this reason, we could not make any Domain Testing.

## Syntax Testing

For syntax testing we use two tools. Because the application is web based, we test using W3C the markup validity for HTML code [5] and CSS co-de validation [6].

Testing with W3C HTML code validation tool we found that www.planillaplus.com code has eight code errors. These errors refer to bad usage of HTML meta tags <meta />, old scripts tag usage that is not valid anymore in HTML v5, and recommendation warnings across the code.

Testing with W3C CSS code validation tool found that www.planillaplus.com CSS code has four errors. These errors refer to parsing errors, and missing semicolon before a property name.

## Finite State Testing

Planilla Plus+ is a web based application. Therefore, no installation is needed to use it. The only requirement is to use any internet browser. The web site is made using HTML5 and PHP code. This means that any modern bowser can browse and navigate the application.

At first glance of the application is not menu driven. This means that you cannot navigate the application using a keyboard only.

The information of the application is stored on a web based databased. The data is saved using a personal account bound with the email. All the progress across the application is saved

automatically. That means all changes or inputs can be saved and continue later.

The application has documented and explained in the frequent question section all common errors. One of the errors we found is that the application let you save the tax forms with errors in the inputs. However, the application shows you all the forms previously completed that have errors and needed to be fixed in the last step. Each error gives you a hyperlink to redirect you where the error is located.

To test the bowser functionality with the application, was performed a capability testing with different browsers to observe their behavior. The browser tested were Safari 10, Google Chrome 54, Internet Explorer 11, and Microsoft Edge. All browsers navigated successfully all web forms of the application.

### Loop Testing

Planilla Plus+ application we can only find non-deterministic loops. Non-deterministic loop is not easily predicted. A loop that is driven by the response of a user is not deterministic, because we cannot predict the response of the user.

We can find all around the application a reusable web form that loops when adding new tax forms. The web form works and looks very simple. It has simple actions to manage like delete, add, edit, and continue. I think is a great idea to manage and keep organize the different tax forms information since adding the loop depend on the type of form you will append to the list.

## CONCLUSION

In this project, we use Black Box testing to the software "Planilla Plus+" using tax form 482 "Planilla Unica" from Puerto Rico. Test results show that any person can complete a tax form 482 but with some issues. One issue is that the application let you fill out and complete a tax form 482 with errors. However, if you fill out correctly all the inputs the tax form will be completed successfully. Black Box testing is very good finding security vulnerabilities and the tests did not show any. The application is oriented to be sequential but sometimes when you have an error you get lost and there is no information why the input is wrong.

The application UI is well made. Looks great and had support for mobile phones and desktop browsers. One of the best highlights of the application is how well manages loops. It reuses a web form on every need of loops to add more tax forms information. Has a great documentation section about the application and possible errors.

After the testing was completed, it is realized that the application has never been tested before. Is always best practice to use a testing methodology to the application? Black Box testing provides a rapid functional testing that can be used in different ways. Normally developers avoid to test, but test helps to find common errors before the clients notice and report. Is recommended to use a test like Black Box testing that focuses on the functionality of the application in collaboration of White Box testing to test the internal structures or workings of the application. Both test are very effective discovering vulnerabilities, validation, implement-tation, functionality, and working issues.

## REFERENCES


[1] "ISTQB exam certification," in *ISTQB*. [Online]. Available: http://www.istqbexamcertification.com/what-is-a-software-testing/. [Accessed: Aug 23, 2016].

[2] M. Pezzè and M. Young, *Software testing and analysis: Process, principles, and techniques.* 1st ed. New York, NY, United States: John Wiley & Sons, 2005.

[3] S. T. Consultans, "Servicios de preparación de planillas en Puerto Rico," in *Planilla Plus*, 2005. [Online]. Available: https://www.planillasplus.com. [Accessed: Aug. 16, 2016].

[4] B. Beizer, *Black-box testing: Techniques for functional testing of software and systems*, 1st ed. New York, NY, United States: John Wiley & Sons, 1995.

[5] "The W3C markup validation service," in *W3C*, 1994. [Online]. Available: http://validator.w3.org/. [Accessed: Dec 1, 2016].

[6] "CSS validation service," in *W3C* CSS, 1994. [Online]. Available: http://www.css-validator.org/. [Accessed: Dec. 1, 2016].