

Application of Base Path for Optimal Testing of Software

Harson Kapoh, Olga Melo, Anthon Arie Kimbal

Dept. Informatics Engineering
State Polytechnic of Manado Indonesia

ABSTRACT

Many purposes for system testing such as testing are carried out to ensure there are no bugs in the application. Tests are also carried out due to the need for a process to evaluate a system both manually and automatically. Testing is carried out by researchers on a software that uses HTML and PHP programming languages. The software developed will be used to connect between the ordering and producers of coconut flour. Software like this is expected to make it easier for customers to place an order by simply accessing orders through the existing application system facilities through the internet network. The application of the testing carried out on the software uses the Base Path Testing method ... Base path Testing is a structural testing group. The process of basic path testing will be considered optimal if all paths that are considered independent in the program, the interface is also a logical decision that has been executed at least once. This test implementation is a white-box testing technique where graph theory is used to analyze the complexity of a structured system. The method that will be used is to make a graph of the control flow of each program function so that optimal testing will be obtained based on the test design. The basic path method is used because of the awareness that there are functions in the system that may not interact with each other. With this testing strategy is expected to help detect the complexity of the structure of the functions developed in the software so that it can be understood by characterizing the relationship, interaction and behavior of the system or software being observed. For this reason, in this article the problem to be answered is how to implement testing using Basic Path Method on ordering coconut flour software. The goal is to produce optimal software by measuring the logical complexity of all execution paths using the basic path method.

General Terms

Testing

Keywords

Basic path test, flow graph, cyclomatic complexity, optimal

1. INTRODUCTION

The software development process usually starts with the software system requirements analysis. During the software development process testing is the core of the software development life cycle [1].

Testing is needed when the software development process and at the end of the development activities. Many purposes for system testing such as testing are carried out to ensure there are no bugs in the application. The tests are also carried out due to the need for a process to evaluate a system both manually and automatically [2]. So that it can be said that software testing is an important task or activity when developing software to ensure that the software meets the requirements desired or determined by the user [3].

There is currently a software that is being developed by researchers using HTML and PHP programming languages. The software developed will be used to connect between the ordering and producers of coconut flour. Software like this is expected to make it easier for customers to place an order by simply accessing orders through the existing application system facilities through the internet network.

The application of the testing carried out on the software uses the Base Path Testing method .. Base path Testing is a structural testing group [4]. The process of basic path testing will be considered optimal if all paths that are considered independent in the program, the interface is also a logical decision that has been executed at least once [5]. This test implementation is a white-box testing technique where graph theory is used to analyze the complexity of a structured system [6].

The method that will be used is to make a graph of the control flow of each program function so that optimal testing will be obtained based on the test design [6].

The basic path method is used because of the awareness that there are functions in the system that may not interact with each other. With this testing strategy is expected to help detect the complexity of the structure of the functions developed in the software so that it can be understood by characterizing the relationship, interaction and behavior of the system or software being observed.

For this reason, in this article the problem to be answered is how to implement testing using Basic Path Method on ordering coconut flour software. The goal is to produce optimal software by measuring the logical complexity of all execution paths using the basic path method.

2. RELATED RESEARCH

Several related studies

The Study of Optimization and Prioritization of Paths in Base Path Testing, this study conducts structural testing in which the tests are compared with other tests such as Genetic algorithm, Ant Colony optimization, Artificial Bee optimization algorithm, Differential Evaluation. This test has the same goal of increasing the effectiveness of the test path based [1].

An efficient method for generating feasible paths for base path testing, this research is carried out and produces software that is suitable for use in the world of routing using the base path method [4].

Functional and Base Path Method for Testing on the Development of the WEB Show Window of Crysanthemum Flowers is a research to test software with the result that software can run efficiently using the base path method test method [5].

Base Path Analysis for Complex System of Systems Testing is a research by implementing the basic path method

technique to test the system with the result that the independent path, all logical decisions, and all interfaces can be run once [6].

3. TEORY

3.1 Basis Path Method

Thomas McCabe introduced a test called Base Path Testing where this test was a white-box test. The implementation is used in procedural design of a system or software. Then with the help of base path testing can be used to help understand and define the size of the logical complexity of existing software. For the use of this path testing basis, the examiner can design a basic set of execution paths that can be carried out to execute each statement in the program with the provisions of at least once during the testing phase [7].

3.2 Flow Graph Notation

Flow graph is widely used in structural testing and consists of a set of N from Nodes and a set of $E \leq N \times N$. To analyze the software there are several notations used: circles are defined as constituent systems, in graphs called nodes. A predicate node is a node that contains conditional capabilities. Individual predicate nodes are needed for each decision point containing one or more Boolean operators. The predicate node must have two sides that are used to exit where one side is for true evaluation results and the other for evaluation that is false. Nodes must be connected to each other represented by arrows as control streams that point towards a specific node or node. The arrow will start and end on a knot and should not intersect and also cross the other edge. Region is an area bounded by a set of edges and nodes. To determine the total area, the examiner must enter the area outside the graph as a region.

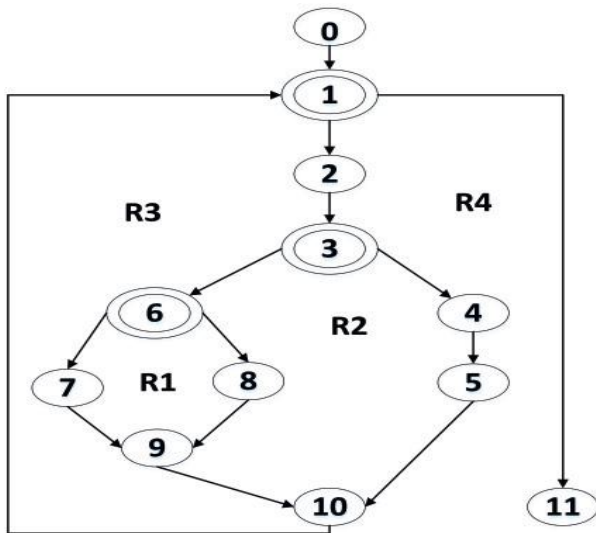


Fig 1: Control flow graph

3.3 Independent System Paths

An independent system path is a path that exists on a graph that can be started from an initial node to another node or end node through several possible paths by following a unique combination of existing nodes and edges.

The base set of independent systems paths for the sample graph above would be the following:

- Path 1: 0-1-11
- Path 2: 0-1-2-3-4-5-10-11
- Path 3: 0-1-2-3-6-8-9-10-11

- Path 4: 0-1-2-3-6-7-9-10-11

3.4 Cyclomatic Complexity

Cyclomatic complexity or conditional complexity is a quantitative measure of the logical complexity of software metrics derived from functional units and is in a basic set defined by the number of independent system paths.

Software metrics that provide a quantitative measure of the logical complexity of the functional unit and that are defined by the number of independent system paths in the base set [8]. Cyclomatic complexity will help provide an upper limit for the number of tests that must be carried out and to make sure that all functions that may have been triggered at least once.

The cyclomatic complexity of a structured SoS (representable by a control flow graph), is defined as:

$$V(G) = E - N + 2 \quad (1)$$

Where (1), E is the number of edges, N is the number of nodes, and P is the number of connected components (exit nodes of the SoS).

If only 1 number of components is connected and goes back to the initial node, then it can be said that the cyclomatic complexity of the entire system of systems matches the cyclical complexity of the graph of the control flow itself and is formulated with $CC = E - N + 2P$, where P is 1 for one SoS system.

For example in the example graph, the Cyclomatic complexity would be computed as follows:

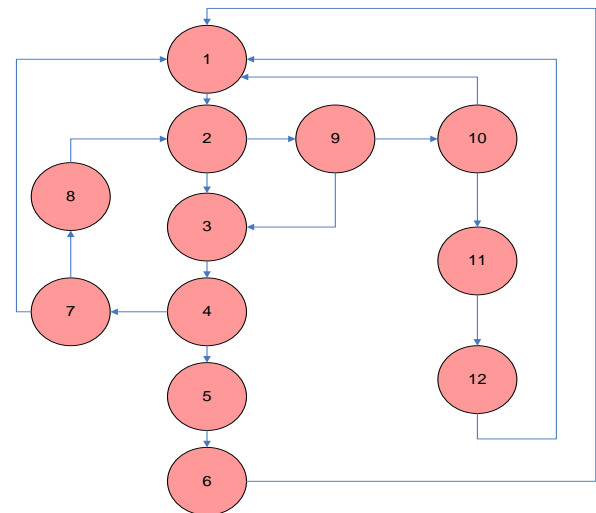


Fig 2: Example control flow graph

The result of the calculation is $16 - 12 + 2 = 5$

After calculating the Cyclomatic complexity and the total number of independent system paths can be determined.

4. METODE

The method used in this study is a software system development method and testing method.

4.1 System Development Methods

The method used in the development of the system is the waterfall method where this method is used to analyze system requirements that support the development of a coconut flour ordering system. Based on the results of the system needs

analysis, the system design is done using several tools such as data flow diagrams for designing data flows, use cases to design software system functionality.

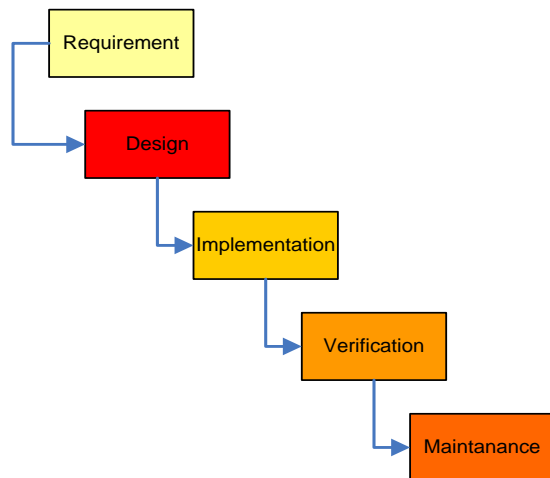


Fig 3: Waterfall

4.2 Testing Methode

The general methodology for testing is as follows [8]:

- Using the System design of the system, an appropriate flow chart image.
- Calculate the Cyclomatic complexity of the flow graph.
- Calculate a basic set of independent system paths.
- Calculate the set of test cases that will cross each path in the specified base.

5. RESULT

5.1 Display of the system

Some system views result from the development as below:

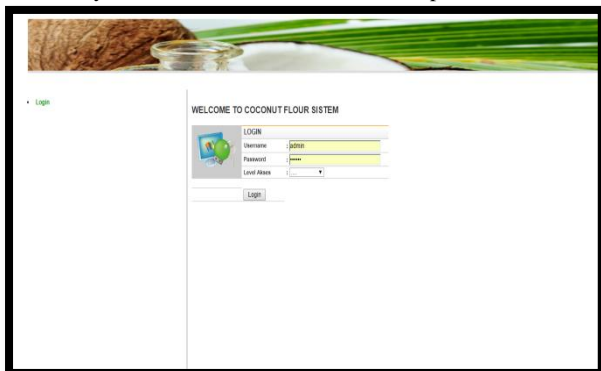


Fig 4: Software login

Fig 5: Form order

Fig 5: Bill

5.2 Application Of Basis Path Testing To An Architecture

The implementation of testing is on the system run by the admin that is filling in customer data.

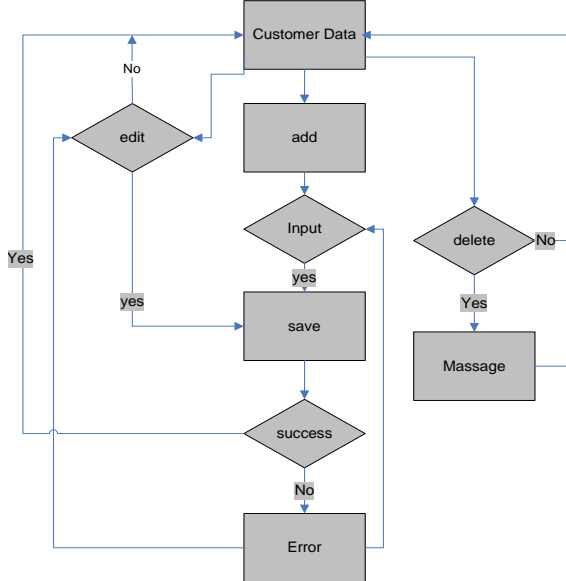


Fig 6: Flowchart

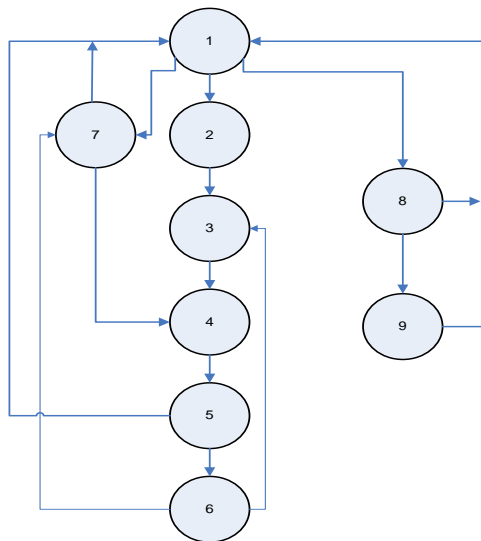


Fig 6: Flow graph

Calculation of Cyclomatic Complexity using the formula

$$V(G) = E - N + 2 \quad (1)$$

The formula (1) is the formula used to determine the logical complexity of the procedure. The sum of E = edge is 14, the sum of N = nodes is 9.

The result of the calculation is $14 - 9 + 2 = 7$

This means that the maximum number of paths on the customer data fill is 7 independent lines and matches the design for test execution that all statements have been executed as below,

Path 1 : 1,7,1

Path 2 : 1,7,4,5,1

Path 3 : 1,7,4,5,6,7,1

Path 4 : 1,7,4,5,6,7,4,5,1

Path 5 : 1,2,3,4,5,6,3,4,5,1

Path 6 : 1,8,1

Path 7 : 1,8,9,1

6. CONCLUSION

Pada pengujian ini kami menemukan kompleksitas penemuan jalur independent oleh karena itu dengan menggunakan metode basis path untuk pengujian white-box didapat rangkaian uji optimal yang menjamin semua jalur independent yang mungkin, semua antar muka dapat dieksekusi setidaknya sekali.

7. ACKNOWLEDGMENTS

Thank you to the leadership and research center of Manado State Polytechnic.

8. REFERENCES

- [1] Gaurav Siwach, Sunil Sikka and Priyanka Makkar. Article: Study of Optimization and Prioritization of Paths in Basis Path Testing. IJCA Proceedings on Innovations in Computing and Information Technology (Cognition 2015) COGNITION 2015(4):6-10, July 2015.
- [2] Rajendrani Mukherjee, K. Sridhar Patnaik, 2018, A survey on different approaches for software test case prioritization, Journal of King Saud University - Computer and Information Sciences, ISSN 1319-1578, <https://doi.org/10.1016/j.jksuci.2018.09.005>.
- [3] Mundita Awotar, Roopesh Kevin Sungkur, Optimization of Software Testing, Procedia Computer Science, Volume 132, 2018, Pages 1804-1814, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2018.05.142>.
- [4] Yan, J. and Zhang, J. (2008). An efficient method to generate feasible paths for basis path testing. Information Processing Letters, 107(3-4), 87-92. doi:10.1016/j.ipl.2008.01.007
- [5] Olga, M. Anthon, A. K. and Robby, T. Functional and Basis Path Method for Testing on the Development of WEB Show Window of Crysanthemum Flowers. International Journal of Computer Applications 175(8):1-6, October 2017.
- [6] Francisco, Z. Aditya, A. Ricardo, P., Eric, S. Basis Path Analysis for Testing Complex System of Systems, Procedia Computer Science, Volume 20, 2013, Pages 256-261, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2013.09.270>.
- [7] T. McCabe. "A complexity Metric", IEEE Trans. Software Engineering, Vol. 2, No. 4, 1976, pp. 308-320
- [8] A. Watson and T. McCabe, "A Testing Methodology Using The Cyclomatic Complexity Matric", Structured Testing: National Institute of Standards and Technology (NIST), Special Publication 500-235, September, 1996.