Home

4 Weeks Away from New Body

and Shape Best Form to Your Body

the best VPN deals

**Best VPNs For** 

**Amazon Fire** 

TV Stick

**Testing** 

**Tutorials** 

White Box Testing

**Cyclomatic Testing** 

**Static Testing** 

**System Testing** 

Code Coverage

Wi Zink

O teu banco fácil

20 sit ups 5 push ups

25 lunges

40 sec plank

25 sec woll sit 10 butt kicks

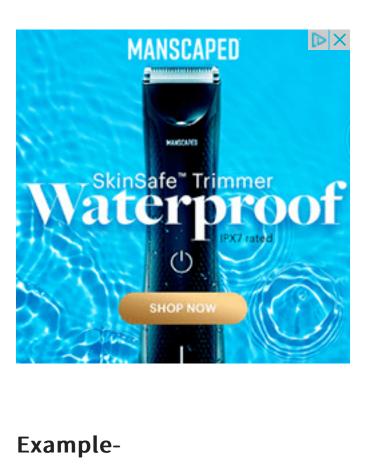
**OFFICE WORKERS** 

Personalized Meal Plan Help to Gain Health, feel Better

FREELANCER

Open

# Mccabe's Cyclomatic Complexity: Calculate with Flow Graph (Example)



first understand -What is Software Metric?

To understand Cyclomatic Complexity, lets

Measurement is nothing but quantitative indication of size / dimension / capacity of an attribute of a product / process. Software metric is defined as a quantitative measure of an attribute a software system possesses with respect to Cost, Quality, Size and Schedule.

## Measure - No. of Errors

In this tutorial, you will learn-• What is Software Metric?

What is Cyclomatic Complexity?

- How to Calculate Cyclomatic Complexity
- Flow graph notation for a program:

Metrics - No. of Errors found per person

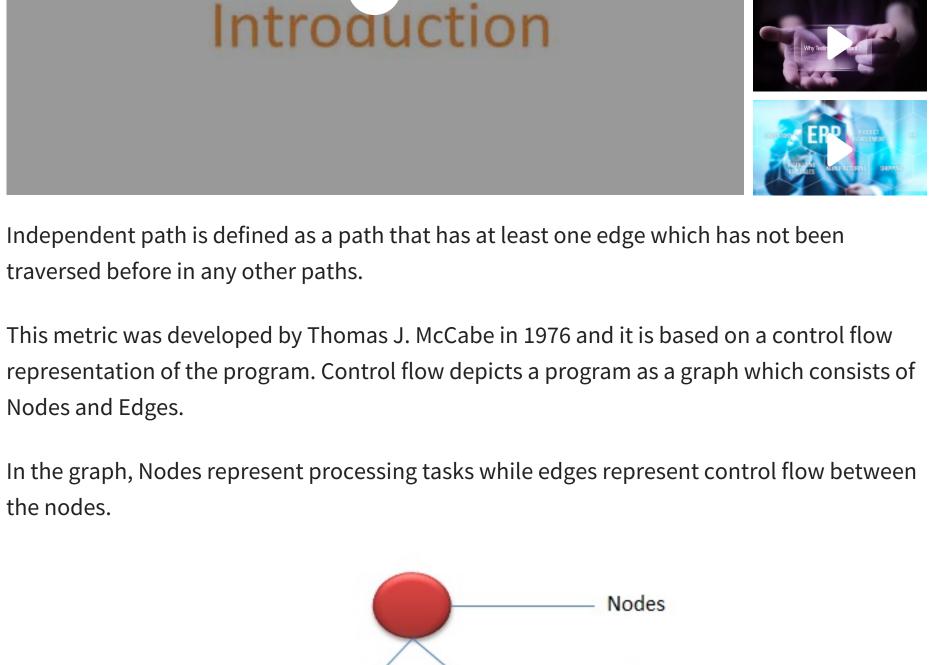
- Properties of Cyclomatic complexity: How this metric is useful for software testing? • More on V (G):
- Tools for Cyclomatic Complexity calculation: • Uses of Cyclomatic Complexity:
- **Cyclomatic Complexity in Software Testing**
- Cyclomatic Complexity in Software Testing is a testing metric used for measuring the
- complexity of a software program. It is a quantitative measure of independent paths in the source code of a software program. Cyclomatic complexity can be calculated by using

software program.

#### **FEATURED VIDEOS** What is Linux Linux Beginner Tutorial

Linux Ovtorials IrRLAYINGn **Linux Tutorials** 

control flow graphs or with respect to functions, modules, methods or classes within a



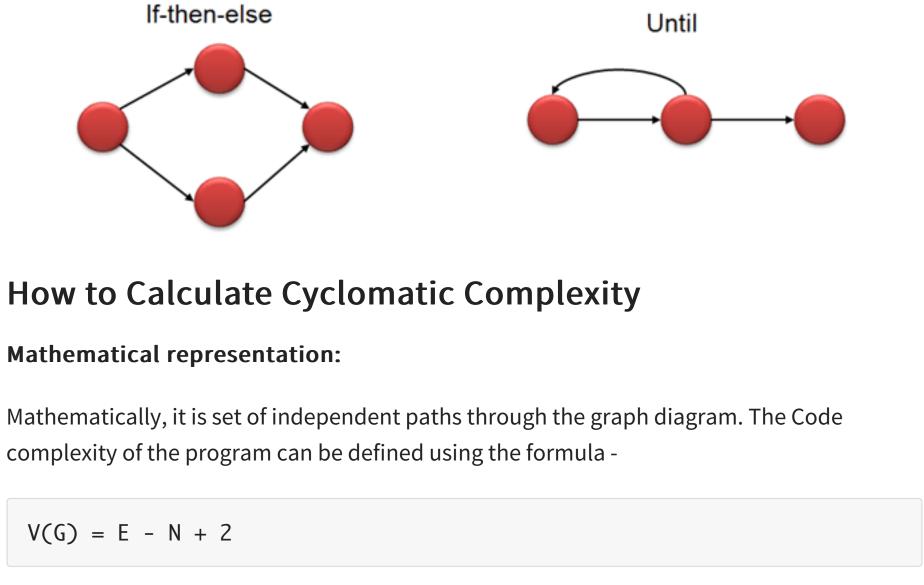
Edges

Flow graph notation for a program: Flow Graph notation for a program defines several nodes connected through the edges.

flow. While

Sequence

Below are Flow diagrams for statements like if-else, While, until and normal sequence of



# V (G) = P + 1

Where P = Number of predicate nodes (node that contains condition)

#### i = 0;n=4; //N-Number of nodes present in the graph

Example -

Where,

E - Number of edges

N - Number of Nodes

while (i<n-1) do j = i + 1;

while (j<n) do

if A[i]<A[j] then

swap(A[i], A[j]);

end do; i=i+1;

### Flow graph for this program will be

end do;

• V(G) = 3 + 1 = 4 (Condition nodes are 1,2 and 3 nodes)

• Basis Set - A set of possible execution path of a program

1. V (G) is the maximum number of independent paths in the graph

Basis Path testing is one of White box technique and it guarantees to execute atleast one



Computing mathematically,

• V(G) = 9 - 7 + 2 = 4

• 1, 2, 3, 4, 5, 2, 6, 1, 7

• 1,7

• 1, 2, 6, 1, 7

2. V(G) >= 1

4. Minimize complexity to 10 How this metric is useful for software testing?

Consider this example -

If (Condition 1)

If (Condition 2)

coverage for the above example.

Steps to be followed:

**Step 4** - Design of Test Cases

More on V (G):

1-10

10-20

>40

cases design.

Statement 1

Statement 2

Else

3. G will have one path if V(G) = 1

statement during testing. It checks each linearly independent path through the program, which means number test cases, will be equivalent to the cyclomatic complexity of the program.

This metric is useful because of properties of Cyclomatic complexity (M) -

2. M can be number of paths through the graphs. (Lower Bound)

1. M can be number of test cases to achieve branch coverage (Upper Bound)

Statement 3 Else Statement 4

The following steps should be followed for computing Cyclomatic complexity and test

# Cyclomatic Complexity for this program will be 8-7+2=3.

**Step 1** - Construction of graph with nodes and edges from the code **Step 2** - Identification of independent paths **Step 3** - Cyclomatic Complexity Calculation

Once the basic set is formed, TEST CASES should be written to execute all the paths.

# (G): **Complexity Number**

**Medium Testability** Cost and effort is Medium Very complex Code 20-40

**Tools for Cyclomatic Complexity calculation:** 

do, catch, case statements in a source code.

Many tools are available for determining the complexity of the application. Some

complexity calculation tools are used for specific technologies. Complexity can be found by

the number of decision points in a program. The decision points are if, for, for-each, while,

Meaning

Complex Code

Low Testability

Not at all testable

Cost and Effort are high

Very high Cost and Effort

Examples of tools are • OCLint - Static code analyzer for C and Related Languages • Reflector Add In - Code metrics for .NET assemblies • GMetrics - Find metrics in Java related applications **Uses of Cyclomatic Complexity:** Cyclomatic Complexity can prove to be very helpful in • Helps developers and testers to determine independent path executions Developers can assure that all the paths have been tested atleast once • Helps us to focus more on the uncovered paths • Improve code coverage in Software Engineering • Evaluate the risk associated with the application or program Using these metrics early in the cycle reduces more risk of the program **Conclusion:** Cyclomatic Complexity is software metric useful for structured or White Box Testing. It is mainly used to evaluate complexity of a program. If the decision points are more, then complexity of the program is more. If program has high complexity number, then probability of error is high with increased time for maintenance and trouble shoot.

#### **Pilot** evels **Testing?** Definition, Meaning, **Orthogonal Array Testing Examples**

What is

**SOFTWARE TESTING** 

(OATS)? Tools,

**Techniques & Example** 

Orthogonal Array Testing

**Orthogonal Array Testing** 

(OAT) is software testing

programming (also called

FP) is a way of thinking

technique that uses...

What is

Prev

YOU MIGHT LIKE:

**SOFTWARE TESTING** 

What is Pilot Testing? PILOT

TESTING is defined as a type

of Software Testing that

and Answers that...

Read more »

**About** 

About Us

Write For Us

**Contact Us** 

**Career Suggestion** 

SAP Career Suggestion Tool

Software Testing as a Career

verifies a...

Read more »

Read more » **SOFTWARE TESTING SDLC** What is **Top 150** 

**Software** 

**Testing Interview Questions Functional** and Answers **Programming? Tutorial** We have compiled the most with Example frequently asked Manual What is Functional **Testing Interview Questions** Programming? Functional

f y in D Advertise with Us Selenium

about...

Read more »

**Testing** 

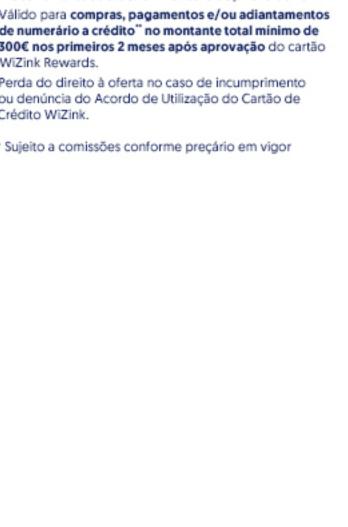
Java

Informatica



Compras online. Em segurança.







As complexity has calculated as 3, three test cases are necessary to the complete path

Cyclomatic complexity can be calculated manually if the program is small. Automated tools need to be used if the program is very complex as this involves more flow graphs. Based on complexity number, team can conclude on the actions that need to be taken for measure. Following table gives overview on the complexity number and corresponding meaning of v Structured and well written code **High Testability** Cost and Effort is less

Deep Behavioural Profiling & Semantic **OPEN** Webhooks **Report a Bug** 

> How to **Write Test** Cases: **Sample Template** with Examples What is a Test Case? A TEST CASE is a set of actions

Next >

**TEST MANAGEMENT** 

**Software Testing** 

(Download Now)

202+ pages eBook...

**SOFTWARE TESTING** 

executed to verify a

Read more »

particular feature or...

Read more »

DY REGISTERED?

**Management PDF for** 

\$20.20 \$9.99 for today 4.5

(114 ratings) Key Highlights

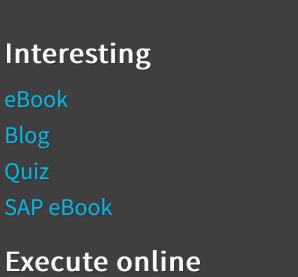
of TEST Management PDF

**Top Tutorials** 



Hacking

**JIRA** 



Execute Java Online

**Execute Javascript** 

© Copyright - Guru99 2021

Execute HTML

**Execute Python** 



**Jmeter** 

Privacy Policy | Affiliate Disclaimer | ToS





