# Homework #5 CSE 566: Software Project/ Process/ Quality Management (2015 spring)

Submitted By: Madhu Meghana Talasila (1207740881)

### **Part 1:** Analyzing quality of software using Resource Standard Metrics Tool

Complexity Metrics, Cyclomatic, Interface, Total (HTML): This metric deals with functional metric and Complexity Analysis

	ers\mtalasil\De	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,			
Function: St					
Parameters:	(Node f, Node	1)			_
Complexity	Param 2	Return 1	Cyclo Vg 1 Comment 0	Total	4
LOC 5	eLOC 4	TLUC 3	Comment 0	Lines	5
Function: St	tack.Stack				
Parameters:	O				
Complexity	Param 0	Return 1	Cyclo Vg 1 Comment 0	Total	2
LOC 3	eLOC 2	lloc 1	Comment 0	Lines	3
Function: St	tack.push				
	(Object data)				
Complexity	Param 1	Return 1	Cyclo Vg 2 Comment 0	Total	4
LOC 9	eLOC 6	1LOC 3	Comment 0	Lines	10
Function: St	tack.non				
Parameters:					
Complexity	Param 0	Return 2	Cyclo Vg 3	Total	5
LOC 14	eLOC 10	1LOC 6	Cyclo Vg 3 Comment 0	Lines	14
<u>Function</u> : Si Parameters:					
		Return 2	Cyclo Vg 2	Total	4
LOC 7	eLOC 5	1LOC 3	Cyclo Vg 2 Comment 0	Lines	7
Function: St	tack.main (String[] args				
Complexity	Param 1	Return 1	Cyclo Ve 1	Total	3
LOC 7	Param 1 eLOC 6	1LOC 5	Cyclo Vg 1 Comment 0	Total Lines	7
	tack.Node.Node				
	(Object d, Nod		Cyclo Ve 1	Total	4
LOC 4	eLOC 3	1LOC 2	Cyclo Vg 1 Comment 0	Lines	4
	~~		e Summary ~~		
	eLOC 42	Total Fil		Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
	eLOC 42	Total Fil	comment 0	Lines	66
File Function Total Funct: Total Funct: Total Funct: Total Cyclo Max Function	eLOC 42	7 Total Fil 1LOC 27  Tile Functi 7 49 36 23 6 11 14 10 6 2 2 4 3	e Summary ~~ Comment 0	Lines  S LOC: S eLOC: S 1LOC: turn .: mplex.: LOC .: eLOC .: lLOC .: meters: rns: polex.:	66
File Function Total Funct: Total Funct: Total Funct: Total Cyclo Max Function	eLOC 42  on Count: ion LOC: ion Params .: Complexity : n LOC: n ELOC: n Parameters: n Returns: c Complex .: tic Complex .:	7 Total Fil 1LOC 27  Tile Functi 7 49 36 23 6 11 14 10 6 2 2 4 3	Comment 0  Conal Summary ~~  Total Function Pt: Total Function Pt: Total Function Pt: Total Function Ret Total Function Ret Total Function Con Average Function Average Function Average Function Return Avg Function Return Avg Interface Company Return Avg Cyclomatic Company Constitution Return Avg Cyclomatic Company Constitution Return Avg Cyclomatic Company Cyclomatic	Lines  S LOC: S eLOC: S 1LOC: turn .: mplex.: LOC .: eLOC .: lLOC .: meters: rns: polex.:	66
File Function Total Funct: Total Funct: Total Funct: Total Funct: Total Cyclo Max Function	eLOC 42  on Count: ion LOC: ion Params .: Complexity : n LOC: n ELOC: n Parameters: n Returns: c Complex .: tic Complex .:	7 Total Fil 1LOC 27 29 36 23 6 11 14 10 6 2 2 4 3 5	Comment 0  Conal Summary ~~  Total Function Pt: Total Function Pt: Total Function Ret Total Function Con Average Function Can Average Function Com Average F	Lines  S LOC: S eLOC: S 1LOC: turn .: mplex.: LOC .: eLOC .: lLOC .: meters: rns: polex.:	66
File Function Total Funct: Total Funct: Total Funct: Total Funct: Total Cyclo Max Function	eLOC 42  con Count: ion LOC: ion Params : Complexity : n LOC: n eLOC: n Parameters: n Returns .: ce Complex : tic Complex : complexity	7 Total Fil 1LOC 27 29 36 23 6 11 14 10 6 2 2 4 3 5	Comment 0  Conal Summary ~~  Total Function Pt: Total Function Pt: Total Function Ret Total Function Con Average Function Can Average Function Com Average F	Lines  S LOC: S eLOC: S 1LOC: turn .: mplex.: LOC .: eLOC .: lLOC .: meters: rns: polex.:	66
File Function Total Funct: Total Funct: Total Funct: Total Cyclo Max Function	eLOC 42  con Count: ion LOC: ion Params : Complexity : n LOC: n eLOC: n Parameters: n Returns .: ce Complex : tic Complex : complexity	7 Total Fil 1LOC 27 29 36 23 6 11 14 10 6 2 2 4 3 5	Comment 0  Conal Summary ~~  Total Function Pt: Total Function Pt: Total Function Ret Total Function Con Average Function Can Average Function Com Average F	Lines  S LOC: S eLOC: S 1LOC: turn .: mplex.: LOC .: eLOC .: lLOC .: meters: rns: polex.:	66
File Function Total Funct: Total Funct: Total Funct: Total Cyclo Max Function Max F	eLOC 42  on Count: ion LOC: ion Params : Complexity : n LOC: n eLOC: n Parameters: n Returns: ce Complex : tic Complex : complexity:	7 Total Fil 1LOC 27  Tile Functi 7 49 36 23 6 11	Comment 0  Conal Summary ~~  Total Function Pt: Total Function Pt: Total Function Ret Total Function Con Average Function Can Average Function Com Average F	Lines  s LOC : s eLOC: turn : mplex.: LOC :: eLOC :: lLOC :: meters: rns .: pplex.: mplex.: ity .:	1.1 0.8 0.5 9 26 7.00 5.14 3.29 0.86 1.29 2.14 1.57 3.71
File Function Total Funct: Total Funct: Total Funct: Total Cyclo Max Function Max F	eLOC 42  on Count: ion LOC: ion Params : Complexity : n LOC: n ELOC: n ELOC: n ELOC: n ELOC: n ELOC: ce Complex : tic Complex : tic Complex : tic Complex : ce Ci\Users\mtal	7 Total Fil 1LOC 27  File Functi 7 49 36 23 6 11	Comment 0  Conal Summary ~~  Total Function Pt: Total Function Pt: Total Function Pt: Total Function Con Average Function Con Average Function I A	Lines  s LOC : s eLOC: turn : mplex.: LOC :: eLOC :: lLOC :: meters: rns .: pplex.: mplex.: ity .:	1.1 0.8 0.5 9 26 7.00 5.14 3.29 0.86 1.29 2.14 1.57 3.71
File Function Total Funct: Total Funct: Total Funct: Total Cyclo Max Function Max F	eLOC 42  on Count: ion LOC: ion Params : Complexity : n LOC: n ELOC: n ELOC: n ELOC: n ELOC: n ELOC: ce Complex : tic Complex : tic Complex : tic Complex : ce Ci\Users\mtal	7 Total Fil 1LOC 27  File Functi 7 49 36 23 6 11	Comment 0  Conal Summary ~~  Total Function Pt: Total Function Pt: Total Function Pt: Total Function Ret Total Function Ret Total Function Con Average Function Average Function Average Function Return Avg Function Return Avg Function Return Avg Total Complex:  Cop\Stack.java	Lines  s LOC : s eLOC: turn : mplex.: LOC :: eLOC :: lLOC :: meters: rns .: pplex.: mplex.: ity .:	1.1 0.8 0.5 9 26 7.00 5.14 3.29 0.86 1.29 2.14 1.57 3.71
File Function Total Funct: Total Funct: Total Funct: Total Funct: Total Cyclo Max Function Max F	eLOC 42  "" F  on Count: ion LOC: ion eLOC: ion Params .: Complexity : " LOC: n LOC: n LOC: c e Complex .: c c Complex .: c com	Total Fil 1LOC 27  File Functi 7 49 36 23 6 11 14 10 6 2 2 4 3 5	Comment 0  Conal Summary ~~  Total Function Pt: Total Function Pt: Total Function Pt: Total Function Con Average Function Con Average Function I A	Lines  s LOC: s eLOC: turn :: mplex.: LOC .: eLOC .: eLOC .: meters: rns .: plex.: ity .:	1.1 0.8 0.5 26 7.00 5.14 3.29 0.86 1.29 2.14 1.57 3.71
File Function Total Funct: Total Funct: Total Funct: Total Funct: Total Cyclo  Max Function Max	eLOC 42  "" F on Count: ion LOC: ion Params .: Complexity :	Total Fil 1LOC 27  Tile Functi 7 49 36 23 6 11	Comment 0  Conal Summary ~~  Total Function Pt: Total Function Pt: Total Function Pt: Total Function Ret Total Function Ret Total Function Con Average Function Average Function Average Function Average Function Average Function Avg Function Param Avg Function Retun Avg Interface Comp Avg Cyclomatic Con Avg Total Complex: Cop\Stack.java  For 1 Files ~~	Lines  s LOC: s eLOC: turn :: mplex.: LOC .: eLOC .: eLOC .: meters: rns .: plex.: ity .:	1.1 0.8 0.5 26 7.00 5.14 3.29 0.86 1.29 2.14 1.57 3.71
File Function Total Funct: Total Funct: Total Funct: Total Funct: Total Cyclo Max Function Max F	eLOC 42  on Count: ion LOC: ion Params :: Complexity : n LOC: n eLOC: n eLOC: n eLOC: ce Complex :: ce Complex :: cic Complex ::	Total Fil  1LOC 27  File Functi  7 49 36 23 6 11 14 10 6 2 2 4 3 3 5  Total Proj	Comment 0  Conal Summary ~~  Total Function Pt: Total Function Pt: Total Function Pt: Total Function Ret Total Function Con Average Function Average Function Average Function Avg Function Para Avg Function Para Avg Function Para Avg Function Retur Avg Interface Comp Avg Total Complex: Cop\Stack.java  For 1 Files ~~	Lines  s LOC : s eLOC: turn : mplex: eLOC .: lLOC : meters: rns .: plex.: mplex.: ity .:	1.1 0.8 0.5 26 7.00 5.14 3.29 0.86 1.29 2.14 1.57 3.71
File Function Total Funct: Total Funct: Total Funct: Total Funct: Total Cyclo Max Function Max F	eLOC 42  "" F on Count: ion LOC: ion Params .: Complexity: " LOC: n LOC: n ELOC: n Returns .: ce Complex .: tic Complex.: tic Complex .: ce Complex .: complex .: ce Complex .: complex .: ce Complex .: c	Total Fil  1LOC 27  Tile Functi  7 49 36 23 6 11	Comment 0  Conal Summary ~~  Total Function Pt: Total Function Pt: Total Function Pt: Total Function Ret Total Function Con Average Function Average Function Average Function Average Function Average Function Return Avg Function Param Avg Function Return Avg Interface Comp Avg Total Complex: Cop\Stack.java  For 1 Files ~~	Lines  s LOC: s eLOC: turn :: mplex.: LOC .: eLOC .: eLOC .: meters: rns .: plex.: ity .:	1.1 0.8 0.5 26 7.00 5.14 3.29 0.86 1.29 2.14 1.57 3.71
File Function Total Funct: Total Funct: Total Funct: Total Funct: Total Cyclo Max Function Max F	eLOC 42  on Count: ion LOC: ion Params :: Complexity : n LOC: n eLOC: n eLOC: n eLOC: ce Complex :: ce Complex :: cic Complex ::	Total Fil  1LOC 27  Tile Functi  7 49 36 23 6 11	Comment 0  Conal Summary ~~  Total Function Pt: Total Function Pt: Total Function Pt: Total Function Ret Total Function Ret Total Function Con Average Function Average Function Average Function Average Function Average Function Avg Function Param Avg Function Return Avg Interface Comp Avg Cyclomatic Con Avg Total Complex: Cop\Stack.java  For 1 Files ~~  Comment 0	Lines  s LOC : s eLOC: turn : mplex: eLOC .: lLOC : meters: rns .: plex.: mplex.: ity .:	1.1 0.8 0.5 26 7.00 5.14 3.29 0.86 1.29 2.14 1.57 3.71

	~~ Pr	oject Funct	ional Metrics ~~		
Function: St					
	(Node f, Node		_		
Complexity	Param 2	Return 1	Cyclo Vg 1 Comment 0	Total	4
LOC 5	eLOC 4	1LOC 3	Comment 0	Lines	5
Function: St	tack.Stack				
Parameters:	O				
Complexity	Param 0	Return 1	Cyclo Vg 1	Total	2
LOC 3	eLOC 2	1LOC 1	Comment 0	Lines	3
Function: St	tack much				
Parameters:	(Object data)				
Complexity	Param 1	Return 1	Cyclo Vg 2	Total	4
LOC 9	eLOC 6	1LOC 3	Cyclo Vg 2 Comment 0	Lines	10
Function: St Parameters:					
Complexity	Param 0	Return 2	Cyclo Vg 3	Total	5
LOC 14	eLOC 10	1LOC 6	Cyclo Vg 3 Comment 0	Total Lines	14
Function: St					
Parameters:		Datum 2	Ovelo va a	Total	4
LOC 7	eLOC 5	1LOC 3	Cyclo Vg 2 Comment 0	Lines	7
	2202 3				
Function: St					
Parameters:	(String[] arg	5)			_
Complexity	Param 1	Return 1	Cyclo Vg 1 Comment 0	Total	3
LUC /	eroc e	ILUC 5	Comment 0	Lines	,
Function: St	tack.Node.Node				
Parameters:	(Object d. No	de n)			
Complexity	Param 2	Return 1	Cyclo Vg 1 Comment 0	Total	4
LOC 4	eLOC 3	1LOC 2	Comment 0	Lines	4
Total: Funct	tions				
	eLOC 36	1LOC 23	InCmp 15	CvcloCmp	11
			_		
Function Po:	ints F	P(LOC) 0.9	FP(eLOC) 0.7	FP(lLOC)	0.4
	~~ Pr	oject Funct	ional Analysis ~~		
Total Funct:				nes:	50
Total Funct:				nes: s LOC :	50 0.9
Total Funct: Total LOC . Total eLOC				nes: s LOC : s eLOC:	0.9 0.7
Total Funct: Total LOC . Total eLOC Total LLOC.				nes: s LOC : s eLOC: s lLOC:	0.9 0.7 0.4
Total Funct: Total LOC . Total eLOC Total 1LOC. Total Cyclor				nes: s LOC : s eLOC: s lLOC:	0.9 0.7 0.4 15
Total Funct: Total LOC . Total eLOC . Total 1LOC. Total Cyclor Total Parame				nes: s LOC : s eLOC: s 1LOC: omp: ts:	0.9 0.7 0.4 15 9
Total Funct: Total LOC . Total eLOC Total 1LOC . Total Cyclor Total Param Total Commer		7 49 36 23 11 6	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines	nes: s LOC : s eLOC: s 1LOC: omp: ts:	0.9 0.7 0.4 15
Avg Physical	ions: matic Comp.: eters: nt Lines:	7 49 36 23 11 6 0	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines		0.9 0.7 0.4 15 9
Avg Physical	natic Comp.: eters t Lines	7 49 36 23 11 6 0	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines		0.9 0.7 0.4 15 9 1
Avg Physical Avg LOC Avg lLOC	natic Comp.: eters nt Lines	7 49 36 23 11 6 0  7.14 7.00 3.29	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines	: mp:	0.9 0.7 0.4 15 9 1 5.14 1.57
Avg Physical Avg LOC Avg lLOC Avg Interface	natic Comp. : eters nt Lines	7 49 36 23 11 6 0 7.14 7.00 3.29 2.14	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines Avg eLOC Avg Cyclomatic Co Avg Parameters	mp:	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86
Avg Physical Avg LOC Avg lLOC Avg Interface	natic Comp. : eters I Lines ce Comp	7 49 36 23 11 6 0 7.14 7.00 3.29 2.14	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines	mp:	0.9 0.7 0.4 15 9 1 5.14 1.57
Avg Physical Avg LOC Avg lLOC Avg Interfac Avg Return I	natic Comp. : eters I Lines ce Comp	7 49 36 23 11 6 0 7.14 7.00 3.29 2.14 1.29	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines Avg eLOC Avg Cyclomatic Co Avg Parameters Avg Comment Lines	mp:	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86
Avg Physical Avg LOC Avg ILOC Avg Interfac Avg Return I Max LOC Max eLOC	natic Comp. : eters nt Lines l Lines ce Comp. : Points	7 49 36 23 11 6 0 7.14 7.00 3.29 2.14 1.29	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines Avg eLOC	mp:	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86
Avg Physical Avg LOC Avg lLOC Avg Interfac Avg Return I Max LOC Max eLOC Max Cyclomat	ions	7 49 36 23 11 6 0 7.14 7.00 3.29 2.14 1.29	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines Avg eLOC	mp:	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00
Avg Physical Avg LOC Avg lLOC Avg Interface Avg Return in Max LOC Max eLOC Max Cyclomat Max Paramete	matic Comp. eters  I Lines  ce Comp  Points  tic Comp	7 49 36 23 11 6 9 7.14 7.00 3.29 2.14 1.29	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines  Avg eLOC Avg Cyclomatic Co Avg Parameters Avg Comment Lines  Max lLOC Max Interface Com Max Return Points	mp:	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00
Avg Physical Avg LoC Avg ILoC Avg Interfac Avg Return I  Max LoC Max eLoC Max Cyclomat Max Paramete Max Comment	ions	7 49 36 23 11 6 0 7.14 7.00 3.29 2.14 1.29	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines  Avg eLOC  Avg Cyclomatic Co Avg Parameters  Avg Comment Lines  Max 1LOC  Max Interface Com Max Return Points Max Total Lines	mp :	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00
Avg Physical Avg LOC Avg ILOC Avg Interface Avg Return I Max LOC Max eLOC Max eLOC Max Paramete Max Comment	matic Comp. eters nt Lines l Lines ce Comp. Points tic Comp. ers	7 49 36 23 11 6 9 7.14 7.00 3.29 2.14 1.29	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines  Avg eLOC  Avg Cyclomatic Co Avg Parameters  Avg Comment Lines  Max lLOC  Max Interface Com Max Return Points Max Total Lines	mp :	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00
Avg Physical Avg LOC Avg ILOC Avg Interface Avg Return I Max LOC Max eLOC Max Cyclomat Max Paramete Max Comment Min LOC Min eLOC	natic Comp. : eters  I Lines  ce Comp  Points  tic Comp  tic Comp	7 49 36 23 11 6 9 7.14 7.00 3.29 2.14 1.29 	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines  Avg eLOC Avg Cyclomatic Co Avg Parameters Avg Comment Lines  Max 1LOC Max Interface Com Max Return Points Max Total Lines  Min 1LOC	mp :	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00
Avg Physical Avg LOC Avg ILOC Avg Interface Avg Return I  Max LOC Max eLOC Max Cyclomat Max Paramete Max Comment  Min LOC Min eLOC Min Cyclomat	matic Comp. eters nt Lines l Lines ce Comp. Points tic Comp. tic Comp.	7 49 36 23 11 6 9 7.14 7.00 3.29 2.14 1.29 1.19 3 2 0	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines  Avg eLOC Avg Cyclomatic Co Avg Parameters Avg Comment Lines  Max 1LOC Max Interface Com Max Return Points Max Total Lines  Min 1LOC Min Interface Com	mp :	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00 6 3 2 14
Avg Physical Avg LOC Avg ILOC Avg Interface Avg Return I  Max LOC Max eLOC Max Cyclomat Max Paramete Max Comment  Min LOC Min cyclomat Min Paramete	natic Comp. eters nt Lines l Lines ce Comp. Points Lines Lines Lines Lines	7 49 36 23 11 6 9 7.14 7.09 3.29 2.14 1.29	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines  Avg eLOC  Avg Cyclomatic Co Avg Parameters  Avg Comment Lines  Max ILOC  Max Interface Com Max Return Points Max Total Lines  Min ILOC  Min Interface Com Min Return Points	mp	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00 6 3 2 14
Avg Physical Avg LOC Avg ILOC Avg Interface Avg Return I  Max LOC Max eLOC Max Cyclomat Max Paramete Max Comment  Min LOC Min cyclomat Min Paramete	matic Comp. eters nt Lines l Lines ce Comp. Points tic Comp. tic Comp.	7 49 36 23 11 6 9 7.14 7.09 3.29 2.14 1.29	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines  Avg eLOC Avg Cyclomatic Co Avg Parameters Avg Comment Lines  Max 1LOC Max Interface Com Max Return Points Max Total Lines  Min 1LOC Min Interface Com	mp	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00 6 3 2 14
Avg Physical Avg LOC Avg ILOC Avg Interface Avg Return I Max LOC Max eLOC Max Cyclomat Max Paramete Max Comment Min LOC Min eLOC Min Cyclomat Min Paramete Min Comment	ions  matic Comp. eters nt Lines  Lines  ce Comp. Points  tic Comp. ers Lines  tic Comp. ers	7 49 36 23 11 6 0 7.14 7.00 3.29 2.14 1.29 14 10 3 2 0	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines  Avg eLOC  Avg Cyclomatic Co Avg Parameters  Avg Comment Lines  Max ILOC  Max Interface Com Max Return Points Max Total Lines  Min ILOC  Min Interface Com Min Return Points	p	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00 6 3 2 14
Avg Physical Avg LOC Avg lLOC Avg Interface Avg Return I Max LOC Max eLOC Max Cyclomat Max Parameto Max Comment Min LOC Min eLOC Min Cyclomat Min Parameto Min Cyclomat Min Cyclomat Min Cyclomat Min Cyclomat Min Cyclomat Min Cyclomat Min Comment	ions  matic Comp. eters nt Lines  Lines  ce Comp. Points  tic Comp. ers Lines  tic Comp. ers	7 49 36 23 11 6 9 7.14 7.00 3.29 2.14 1.29 	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines  Avg eLOC  Avg Cyclomatic Co Avg Parameters  Avg Comment Lines  Max 1LOC  Max Interface Com Max Return Points Max Total Lines  Min 1LOC  Min Interface Com Min Return Points Min Total Lines	p	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00 6 3 2 14
Avg Physical Avg LOC Avg ILOC Avg Interface Avg Return I  Max LOC Max eLOC Max Cyclomat Max Paramete Max Comment  Min LOC Min Cyclomat Min Paramete Min Comment	ions  matic Comp. eters nt Lines  Lines  ce Comp. Points  tic Comp. ers Lines  tic Comp. ers	7 49 36 23 11 6 0 7.14 7.00 3.29 2.14 1.29 14 10 3 2 0 3 2 1 0 0	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines  Avg eLOC  Avg Cyclomatic Co Avg Parameters  Avg Comment Lines  Max ILOC  Max Interface Com Max Return Points Max Total Lines  Min ILOC  Min ILOC  Min Return Points Min Total Lines	mp :	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00 6 3 2 14
Avg Physical Avg LOC Avg lLOC Avg Interface Avg Return I  Max LOC Max eLOC Max Cyclomat Max Paramete Max Comment  Min LOC Min cyclomat Min Paramete Min Comment  C Source Fil C++ Source I	ions  matic Comp. eters nt Lines  Lines  ce Comp. Points  Lines  Lines  Lines  Lines  Lines  Lines  Lines  Lines  Lines	7 49 36 23 11 6 9 7.14 7.00 3.29 2.14 1.29 14 10 3 2 0 3 2 1 0 0 0	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines  Avg eLOC  Avg Cyclomatic Co Avg Parameters  Avg Comment Lines  Max ILOC  Max Interface Com Max Return Points Max Total Lines  Min ILOC  Min ILOC  Min Iterface Com Min Return Points Min Total Lines  Gummary ~~  C/C++ Include Files	mp	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00 6 3 2 14
Avg Physical Avg LOC Avg lLOC Avg Interface Avg Return I  Max LOC Max eLOC Max Paramete Max Comment  Min LOC Min eLOC Min Cyclomat Min Paramete Min Comment  C Source Fit C++ Sou	ions  matic Comp. eters nt Lines  Lines  ce Comp. Points  Lines  Lines  Lines  Lines  tic Comp. ers  Lines  Lines  Lines  Lines	7 49 36 23 11 6 9 7.14 7.09 3.29 2.14 1.29 14 10 3 2 0 3 2 1 0 0	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Point Total Blank Lines  Avg eLOC  Avg Cyclomatic Co Avg Parameters  Avg Comment Lines  Max lLoc  Max Interface Com Max Return Points Max Total Lines  Min lLoC  Min Interface Com Min Return Points Min Total Lines  Gummary ~~  C/C++ Include Fil	mp	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00 6 3 2 14
Avg Physical Avg LOC Avg ILOC Avg Interface Avg Return I  Max LOC Max eLOC Max Paramete Max Comment  Min LOC Min Cyclomat Min Paramete Min Comment  C Source Fil C++ Source I C# Source Fil Other Source	ions  matic Comp. eters nt Lines  Lines  ce Comp. Points  Lines  Lines  Lines  Lines  Lines  Lines  Lines  Lines  Lines	7 49 36 23 11 6 9 7.14 7.00 3.29 2.14 1.29 14 10 3 2 0 3 2 1 0 0 0	Total Physical Li Total Function Pt Total Function Pt Total Function Pt Total Function Pt Total Interface C Total Return Poin Total Blank Lines  Avg eLOC  Avg Cyclomatic Co Avg Parameters  Avg Comment Lines  Max ILOC  Max Interface Com Max Return Points Max Total Lines  Min ILOC  Min ILOC  Min Iterface Com Min Return Points Min Total Lines  Gummary ~~  C/C++ Include Files	mp	0.9 0.7 0.4 15 9 1 5.14 1.57 0.86 0.00 6 3 2 14

As observed in the above image,

Comments in this file are Zero. So the program that I used has less maintainability. Also Logical lines of code are less than effective lines of code. There might be empty spaces and braces placed for readability. Cyclomatic[1] complexity is very little so there is less branching. Similarly no. of return points, no. of interface components and minimum, average and maximum eLOC, lLOC and LOC are given in project Functional analysis and File Functional analysis.

Now, we will check the Functional Quality Metrics: It deals with Function Metrics, Class/ Struct Metrics, Complexity Analysis, Quality Analysis.

In this each of the class and function is analyzed in function metrics with their corresponding notices, complexity LOC, eLOC, lLOC, comments, lines etc. As mentioned earlier there are no comments in the file so comment count is shown zero. We can check each of the notice and work to ensure the quality of the program.

Then total functional summary and File functional Summary are given. Which analyses the total no. of modifications needed to be done to make the code effective is given.

Class/ Struct Metrics is the difference in thos Functional Quality Metrics, This defines the no. of access modifiers used in the program. No. of public, private, protected class variables, functions, methods are there in the program. Depending on that we can make the changes in the code and we can try to avoid particular access modifier depending on program need.

Finally a project quality profile and a quality density report are given. Which gives the total number of notices and where modifications are necessary in detail. For suppose file comment content is <10% which is very bad. Another instance, functions appears to have blank or null constraints. Which have a considerable impact on the memory of the program thereby effects the quality of code. Type, count and percent of a quality notice is clearly mentioned in the project quality profile. In project density profile, percentage of LOC, ELOC and ILOC are mentioned.

#### **Functional Quality Metrics**

```
Report Banner - Edit rsm.cfg File
              Resource Standard Metrics™ for C, C++, C# and Java
Version 7.75 - <u>mSquaredTechnologies.com</u>
License Type: Shareware Evaluation License
Licensed To : Shareware End User - Distribute Freely
License No. : SW1380 License
Build Date : Sep 2 2009 Ru
01996-2009 M Squared Technologies LLC<sup>®</sup>
                                                             License Date: Dec 05, 1998
Run Date: Apr 23, 2015
Functional Quality Metrics
                               ~~ Function Metrics ~~
~~ Class/Struct Metrics ~
                                File: C:\Users\mtalasil\Desktop\Stack.java
                   ------ Class Begin Line: 1 ------
Class: Stack
  Notice #52: Line 1: A class has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.
                   ----- Function Begin Line: 5 -----
Function: Stack.Stack
Parameters: (Node f, Node 1)
  Notice #51: Line 5: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.
Function: Stack.Stack
--- Function Begin Line: 11 -----
Function: Stack.Stack
Parameters: ()
  Notice #51: Line 11: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.
  Notice #49: The function contains no input parameters or void.
Suggest using explict parameters for interface clarity.
Function: Stack.Stack
                        0 Return 1 Cyclo Vg 1 Total
2 lLOC 1 Comment 0 Lines
----- Function End Line: 13
Complexity Param 0
LOC 3 eLOC 2
                        ----- Function Begin Line: 15 -----
Function: Stack.push
Parameters: (Object data)
  Notice #51: Line 15: A function has been identified which does not
            have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.
  Notice #17: Function comments, 0.0% are less than 10.0%.
```

```
------ Function Begin Line: 26
Function: Stack.pop
Parameters: ()
  Notice #51: Line 26: A function has been identified which does not
         have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.
  Notice #17: Function comments, 0.0% are less than 10.0%.
  Notice #46: Function blank line percent, 0.0% is less than 10.0%.
  Notice #27: The number of function return points
         2 exceeds the specified limit of 1.
  Notice #49: The function contains no input parameters or void.
         Suggest using explict parameters for interface clarity.
Function: Stack.pop
Complexity Param 0
LOC 14 eLOC 10
                          Return 2
                                       Cyclo Vg 3
                                                          Total
                                        Comment 0
                                                                     14
                          1LOC 6
                                                         Lines
           ----- Function End Line: 39 -----
          ------ Function Begin Line: 41 -----
Function: Stack.peek
Parameters: ()
  Notice #51: Line 41: A function has been identified which does not
         have a preceding comment. Comments that detail the purpose,
         algorithms, and parameter/return definitions are suggested.
  Notice #17: Function comments, 0.0% are less than 10.0%.
  Notice #46: Function blank line percent, 0.0% is less than 10.0%.
  Notice #27: The number of function return points
         2 exceeds the specified limit of 1.
  Notice #49: The function contains no input parameters or void.
         Suggest using explict parameters for interface clarity.
Function: Stack.peek
Complexity Param 0
LOC 7 eLOC 5
                           Return 2
                                        Cyclo Vg 2
                                         Comment 0
                          1LOC 3
                 ------ Function End Line: 47 ------
            ----- Function Begin Line: 49 -----
Function: Stack.main
Parameters: (String[] args)
  Notice #51: Line 49: A function has been identified which does not
         have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.
  Notice #17: Function comments, 0.0% are less than 10.0%.
  Notice #46: Function blank line percent, 0.0% is less than 10.0%.
Function: Stack.main
Complexity Param 1
                                         Cyclo Vg 1
                           Return 1
                                                          Total
100 7
            eLOC 6
                           1LOC 5
                                         Comment 0
                                                          Lines
        ----- Function End Line: 55
       ------ Nested Class Begin Line: 57 ------
Nested Class: Stack.Node
  Notice #52: Line 57: A class has been identified which does not
        have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.
                ------ Function Begin Line: 61 ------
Function: Stack.Node.Node
Parameters: (Object d, Node n)
  Notice #51: Line 61: A function has been identified which does not
        have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.
Function: Stack.Node.Node
Complexity Param 2 Return 1 Cyclo Vg 1
                                                          Total
            eLOC 3
                          1LOC 2
                                         Comment 0
LOC 4
                                                         Lines
------ Function End Line: 64 ---------
```

	s: Stack.Node	Prot 0	Private 2	Total	2
Methods		Prot 0	Private 1	Total	1
LOC 8	eLOC 6	1LOC 4	Comment 0	Lines	8
			End Line: 65		
Notice #31	l: Class commen	ts, 0.0% a	re less than 10.0%		
Class: Stack					
Attributes	Publ 0	Prot 0	Private 2 Private 0	Total Total	2
Methods	Publ 6	Prot 0	Private 0	Total	6
			Comment 0		65
		CIASS LIIU	Eine. oo iiii		
			entage, 0.0% is le		
		Total Fil	e Summary ~~		
		1LOC 27	Comment 0		
	~~ F	ile Functi	onal Summary ~~		
File Function	on Count:	7	Total Function Pt: Total Function Pt: Total Function Re: Total Function Cor Average Function		
Total Functi	ion LOC:	49	Total Function Pt	s LOC :	1.1
Total Functi	ion eLOC:	36	Total Function Pt	s eLOC:	0.8
Total Functi	ion lloc:	23	Total Function Pt	s lloc:	0.5
Total Cyclo	Complexity:	11	Total Function Cor	molex.:	9 26
					20
					7.00
	1 eLOC:	10	Average Function (	eLOC .:	5.14
	1 1LOC:		Average Function :		3.29
May Eunction	Danameters	2	Avg Function Para Avg Function Return Avg Interface Comp Avg Cyclomatic Comp Avg Total Complex:	meters:	0.86
Max Function	Returns: ce Complex. :	2	Avg Function Retu	rns:	1.29
Max Interfac	ce Complex. :	4	Avg Interface Comp	olex. :	2.14
	tic Complex.:	3	Avg Cyclomatic Com	mplex.:	1.57 3.71
Max Total Co	Amplexity	-	Avg local complex	LLY	3.71
End of File:	: C:\Users\mtal	asil\Deskt	op\Stack.java		
			op\Stack.java For 1 Files ~~		
	~~ Tota	l Metrics	For 1 Files ~~		
	~~ Tota	l Metrics	For 1 Files ~~		
LOC 57	~~ Tota	1 Metrics Total Proj	For 1 Files ~~ ect Summary ~~	Lines	66
LOC 57 Average per	~~ Tota ~~. eLOC 42 File, metric/1	l Metrics Total Proj LLOC 27 files	For 1 Files ~~ ect Summary ~~ Comment 0	Lines	66
LOC 57	~~ Tota	1 Metrics Total Proj	For 1 Files ~~ ect Summary ~~		
LOC 57 Average per LOC 57	~~ Tota ~~. eLOC 42 File, metric/1	l Metrics  Total Proj  Loc 27 files Loc 27	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0	Lines	66 66
LOC 57 Average per LOC 57	eLOC 42 File, metric/1 eLOC 42	l Metrics Total Proj lLOC 27 files lLOC 27	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0	Lines Lines	66 66
LOC 57 Average per LOC 57	eLOC 42 File, metric/1 eLOC 42 ~~ Pro	l Metrics  Total Proj  lLOC 27 files lLOC 27	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0	Lines Lines	66 66
LOC 57 Average per LOC 57	eLOC 42 File, metric/1 eLOC 42  ~~ Pro	l Metrics  Total Proj  lLOC 27 files lLOC 27	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0	Lines Lines	66 66
LOC 57 Average per LOC 57 Function: St	eLOC 42 File, metric/1 eLOC 42  ~~ Pro	l Metrics  Total Proj  Loc 27 files Loc 27  ject Funct	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0	Lines Lines	66 66
LOC 57 Average per LOC 57 Function: St Parameters: Complexity LOC 5 Function: St	eLOC 42 File, metric/1 eLOC 42  www Properties of the Control of t	l Metrics  Total Proj  LOC 27 files LOC 27  ject Funct	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0  ional Metrics ~~	Lines Lines	66 66
LOC 57 Average per LOC 57 Function: Serial S	eLOC 42 File, metric/1 eLOC 42  ~~ Pro tack.Stack (Node f, Node) Param 2 eLOC 4 tack.Stack ()	l Metrics  Total Proj  LOC 27 files LOC 27  ject Funct	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0  ional Metrics ~~	Lines Lines	66 66
LOC 57 Average per LOC 57 Function: St Parameters: Complexity LOC 5 Function: St Parameters:	eLOC 42 File, metric/1 eLOC 42  ~~ Pro tack.Stack (Node f, Node) Param 2 eLOC 4 tack.Stack ()	l Metrics  Total Proj  Loc 27 files Loc 27  ject Funct	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0  ional Metrics ~~  Cyclo Vg 1  Comment 0	Lines Lines Total Lines	66 66 
LOC 57 Average per LOC 57 Function: St Parameters: Complexity LOC 5 Function: St Parameters: Complexity LOC 3 Function: St	eLOC 42 File, metric/1 eLOC 42  ~~ Pro tack.Stack (Node f, Node Param 2 eLOC 4 tack.Stack () Param 0 eLOC 2 tack.push	l Metrics  Total Proj  Loc 27 files Loc 27 ject Funct  Return 1 Loc 3	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0  ional Metrics ~~  Cyclo Vg 1  Comment 0	Lines Lines Total Lines	66 66 
LOC 57 Average per LOC 57 Function: St Parameters: Complexity LOC 5 Function: St Parameters: Complexity LOC 3 Function: St Parameters:	eLOC 42 File, metric/1 eLOC 42	l Metrics  Total Proj  LOC 27 files 1LOC 27  ject Funct  Return 1 1LOC 3  Return 1 1LOC 1	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0  ional Metrics ~~  Cyclo Vg 1  Comment 0  Cyclo Vg 1  Comment 0	Lines Lines Total Lines Total Lines	66 66 
LOC 57 Average per LOC 57 Function: St Parameters: Complexity LOC 5 Function: St Parameters: Complexity LOC 3 Function: St	eLOC 42 File, metric/1 eLOC 42	l Metrics  Total Proj  Loc 27 files Loc 27 ject Funct  Return 1 Loc 3	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0  ional Metrics ~~  Cyclo Vg 1  Comment 0	Lines Lines Total Lines	66 66 
LOC 57 Average per LOC 57 Function: St Parameters: Complexity LOC 5 Function: St Parameters: Complexity LOC 3 Function: St Parameters: Complexity LOC 9 Function: St	eLOC 42 File, metric/1 eLOC 42	l Metrics  Total Proj  Loc 27 files Loc 27  ject Funct  Return 1 Loc 3  Return 1 Loc 1	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0  ional Metrics ~~  Cyclo Vg 1  Comment 0  Cyclo Vg 1  Comment 0	Lines Lines Total Lines Total Lines	66 66  4 5
LOC 57 Average per LOC 57 Function: St Parameters: Complexity LOC 5 Function: St Parameters: Complexity LOC 3 Function: St Parameters: Complexity LOC 9 Function: St Parameters: Parameters:	eLOC 42 File, metric/1 eLOC 42	l Metrics  Total Proj  Loc 27 files 1LOC 27  ject Funct  Return 1 1LOC 3  Return 1 1LOC 1	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0  ional Metrics ~~  Cyclo Vg 1  Comment 0  Cyclo Vg 1  Comment 0	Lines Lines Total Lines Total Lines	66 66 4 5
LOC 57 Average per LOC 57 Function: St Parameters: Complexity LOC 5 Function: St Parameters: Complexity LOC 3 Function: St Parameters: Complexity LOC 9 Function: St	eLOC 42 File, metric/1 eLOC 42	l Metrics  Total Proj  Loc 27 files Loc 27  ject Funct  Return 1 Loc 3  Return 1 Loc 1	For 1 Files ~~  ect Summary ~~  Comment 0  Comment 0  ional Metrics ~~  Cyclo Vg 1  Comment 0  Cyclo Vg 1  Comment 0	Lines Lines Total Lines Total Lines	66 66  4 5

Function: St Parameters: Complexity LOC 7	()	Return 2 1LOC 3	Cyclo Vg 2	Total Lines	4 7
Function: St		1100 3	Connect 6	Lines	,
	(String[] arg	s)			
Complexity	Param 1	Return 1	Cyclo Vg 1	Total	3
LOC 7	eLOC 6	1LOC 5	Comment 0	Lines	7
Parameters:	tack.Node.Node (Object d, No	de n)	_		
Complexity	Param 2 eLOC 3	Return 1	Cyclo Vg 1	Total	4
LOC 4	eLOC 3	1LOC 2	Comment 0	Lines	4
Total: Funct		11.05.23	InCmp 15	Cuslaces	
Function Pos	ints F	P(LOC) 0.9	FP(eLOC) 0.7	FP(lLOC)	0.4
	~~ Pr		ional Analysis ~~		
	ions:	7	Total Physical Li	nes:	50
			Total Function Pt Total Function Pt		0.9
			Total Function Pt		0.7 0.4
	matic Comp. :		Total Interface C		15
	eters:		Total Return Poin		9
	nt Lines:	9	Total Blank Lines		1
	l Lines:	7.14	Avg eLOC		E 14
			Avg Cyclomatic Co	mp.	5.14 1.57
Avg Interfac	ce Comp:		Avg Parameters	•	0.86
	Points:		Avg Comment Lines		0.00
		14			_
	tic Comp:		Max 1LOC Max Interface Com		6 3
	ers		Max Return Points		2
	Lines:		Max Total Lines .		14
		3			_
	tic Comp		Min ILOC		1
	tic Comp:		Min Interface Com Min Return Points		1
			Min Total Lines .		3
			Struct Metrics ~~ :lude Nested LOC Me	trics	
Class: Stack	k				
Attributes	Publ 0	Prot 0	Private 2	Total	2
Methods	Publ 6	Prot 0	Private 0	Total	6
Complexity	Param 6	Return 9	Cyclo Vg 11	Total	26
LOC 57	eLOC 42	1LOC 27	Comment 0	Lines	65
Nested Class	s: Stack.Node				
Attributes	Publ 0	Prot 0	Private 2	Total	2
Methods	Publ 0	Prot 0	Private 1	Total	1
Complexity	Param 2	Return 1	Cyclo Vg 1	Total	4
LOC 8	eLOC 6	1LOC 4	Comment 0	Lines	8
Total: All F	Parent Classes	/Structs			
Attributes		Prot 0	Private 2	Total	2
Methods	Publ 6	Prot 0	Private 0	Total	6
Complexity		Return 9	Cyclo Vg 11	Total	26
LOC 57	eLOC 42	1LOC 27	Comment 0	Lines	65

```
~~ Project Class/Struct Analysis ~~
Total Classes/Structs .:
Total Nested Classes .: 1 Total Methods .....:
Total Public Methods .: 6 Total Public Attributes:
Total Protected Methods: 9 Total Protected Attrib.:
Total Private Methods .: 9 Total Private Attrib.:
Total Physical Lines .: 65 Total LOC .....:
Total eLOC ......: 42 Total LOC .....:
Total Cyclomatic Comp.: 11 Total Interface Comp. .:
Total Parameters ....: 6 Total Return Points ...:
Total Comment Lines ...: 9 Total Blank Lines ....:
                                                                                                  2
                                                                                                57
                                                                                                 27
                                     6 Total Interface Comp. .:
6 Total Return Points ...:
9 Total Blank Lines ....:
                                                                                                15
Total Comment Lines ...:
                                                                                                  Q
                        ----
Avg Physical Lines ....:
                                        32.50 Avg Methods ....:
                                                                                              3.50
                                     3.00 Avg Public Attributes .:
0.00 Avg Protected Attrib. .:
0.00 Avg Private Attributes :
Avg Public Methods ....:
Avg Protected Methods .:
Avg Private Methods ...:
                                                                                              1.00
Avg LOC .....
                                        28.50 Avg eLOC ....:
13.50 Avg Cyclomatic Comp. ..:
                                                                                           21.00
                                                                                              5.50
Avg Interface Comp. ...:
Avg Return Points ....:
                                        7.50 Avg Parameters .....:
4.50 Avg Comment Lines ....:
                                                                                             3.00
                                                                                             0.00
                                                                   -----
                        ----
                                       65 Max Methods ....:
Max Physical Lines ....:
Max Public Methods ....:
Max Protected Methods .:
Max Private Methods ...:
                                                                                                  2
42
                                                                                                11
Max Return Points ....:
                                             9 Max Comment Lines ....:
                                                                                                  8
                                        Min Physical Lines ....:
Min Public Methods ....:
Min Protected Methods .:
Min Private Methods ...:
Min LOC ....:
Min Return Points ....:
                              ~~ Project Quality Profile ~~
Type Count Percent Quality Notice
17
                  17.39 Function comment content less than 10.0%
                   4.35 File comment content < 10.0%
20
            1
                   8.70 Number of function return points > 1
27
                    4.35 Class/Struct comments are < 10.0%
31
             1
             3 13.04 Function/Class Blank Line content less < 10.0%
3 13.04 Function appears to have null or blank parameters</pre>
46
49
                 30.43 No comment preceding a function block
8.70 No comment preceding a class block
51
52
             2
            23 100.00 Total Quality Notices
                              ~~ Quality Notice Density ~~
Basis: 1000 (K)
Quality Notices/K LOC = 403.5 ( 40.35%)
Quality Notices/K eLOC = 547.6 ( 54.76%)
Quality Notices/K lLOC = 851.9 ( 85.19%)
                                        -----
                                     ~~ File Summary ~~
C Source Files *.c ...:
C++ Source Files *.c* .:
C# Source Files *.cs ..:
                                            0 C/C++ Include Files *.h:
0 C++ Include Files *.h*:
0 Java Source File *.jav*:
Other Source Files ....:
Total File Count .....:
                                              8
                                              1
```

#### **Code:**

```
public class Stack {
  Node first;
  Node last;
  public Stack(Node f, Node l) {
     first = f;
    last = 1;
     first.next = last;
  }
  public Stack() {
     first.next = last;
  }
  public void push(Object data) {
    if(first == null) {
       first = new Node(data, null);
     }
     else {
       last.next = new Node(data, null);
       last = last.next;
  }
  public Object pop() {
     if(first == null) {
```

```
return -1;
  }
  else {
     Object item = last.data;
     Node cur = first;
     while (cur.next.next != null) {
       cur = cur.next;
     }
     last = cur;
    return item;
}
public Object peek() {
  if(first == null) {
    return -1;
  }
  Object item = last.data;
  return item;
}
public static void main(String[] args) {
  Stack stack = new Stack(new Node(1, null), new Node(2, null));
  stack.push(3);
  System.out.println(stack.peek() == 3);
  stack.pop();
  System.out.println(stack.peek() == 2);
}
```

```
private static class Node {
    Object data;
    Node next;

private Node(Object d, Node n) {
    data = d;
    next = n;
    }
}
```

#### Part 2

The proposed metric is on "Metrics of Software Architecture Changes Based on Structural Distance".

Proposed Software Architecture Metric in "Metrics of Software Architecture Changes Based on Structural Distance": "This paper concentrates on different versions of code and is trying to analyze the architectural change using Structural Distance rather than at architecture metrics like size, complexity, coupling and cohesion. This concentrates on changes between the versions at each individual component level. "In this paper, two endpoints of a major change are taken as reference points, and intermediate connectivity changes are examined relative to the endpoints". They defined a graph kernel function to measure distance between software structures. This measures the architecture change as a transition between two end points [2]".

"How does it relates to RSM tool Metrics?" It is also considering the functions, classes and lines but differently. Instead of finding size, complexity, coupling and soon it is trying to find the Structural Distance between various components in different versions of program and trying to analyze the architectural (i.e., what contents and where the contents are changes in the file physically so that modified there will saves a lot of time and can be done efficiently).

"My Personal Opinion": This metric seemed to be useful as while different versions of the code are created. Most of the structure remains same expects the functions or modules where changes are required. Tracking those changes based on number of lines of code or functionality of the code or coupling between the codes seems to be less effective. For suppose while comparing two triangles if we compare distance between the edges then there is high probability of accuracy than comparing triangles for number of points or number of sides. Also, Working at module level seems to be very easy for a programmer to understand the changes than at line level or number of lines. I feel this metric is useful while comparing different versions of software rather than using RSM.

"Validation of Proposed Metric": Used on four open source projects which used java as their source code on different versions of four projects. Then considered class as nodes and if any changes to nodes are not considered as, even if the names of the classes are changed software architecture won't be effected. But this is not possible with lines of code metric as change of class name is also considered as change of line or addition of a space also a possible change even though which has less impact on architectural change. Then they captured properties like inheritance, association and user dependencies which have major impact on the code. Then analyzed the metric on different versions of open source code and presented the graphical analysis of transition between two architectures. Then they compared their metric to Lines of Code. Using Lines of Code they do observed continuous increase which does not mean no significant changes in code or Lines of Code failed to capture the changes properly. This metric clearly validated the Software Architecture metric which enhances the quality of the program.

## References

- $1. \ http://msquared technologies.com/m2 rsm/docs/index.htm$
- $2. \ https://www.cs.umd.edu/{\sim} basili/publications/proceedings/P114.pdf$