

การวัดความซับซ้อนของซอฟต์แวร์

SOFTWARE COMPLEXITY MEASUREMENT



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาค้นคว้าตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2544

ISBN 974-646-326-5

การวัดความซับซ้อนของซอฟต์แวร์
SOFTWARE COMPLEXITY MEASUREMENT



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2544

ISBN 974-648-328-5

เลขหมู่.....
เลขทะเบียน 40380
วัน, เดือน, ปี 16 ก.ย. 2544

b.....

ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOFTWARE COMPLEXITY MEASUREMENT



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2001

ISBN 974-648-328-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2001

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การวัดความซับซ้อนของซอฟต์แวร์
SOFTWARE COMPLEXITY MEASUREMENT

ชื่อนักศึกษา นางสาวนิภาพร ประภาศิริ

รหัสประจำตัว 38626037

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา เทคโนโลยีสารสนเทศ

อาจารย์ผู้ควบคุมวิทยานิพนธ์ ผศ.ดร.เอื้อน ปิ่นเงิน

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.เอื้อน	ปิ่นเงิน	อ.ไพรัช
รศ.ดร.วิเชียร	เปรมชัยสวัสดิ์	
รศ.ดร.นุชรี	เปรมชัยสวัสดิ์	
ดร.ประจวบ	วานิชชัชวาล	
ดร.ชุติเมษณ์	ศรีนิตทา	ทุทเพชญ์ ปิ่นเงิน

วัน/เดือน/ปี ที่สอบ 13 มิถุนายน 2544 เวลา 13.00 น. เป็นต้นไป

สถานที่สอบ ห้องบรรยาย 234-235 คณะเทคโนโลยีสารสนเทศ สจล.

บัณฑิตวิทยาลัยรับรองแล้ว



(รศ.ดร.บุญวัฒน์ อัคร)

คณบดีบัณฑิตวิทยาลัย

วันที่.....๑๖.....เดือน.....พ.ศ.....๒๕๔๔

หัวข้อวิทยานิพนธ์	การวัดความซับซ้อนของซอฟต์แวร์
นักศึกษา	นางสาวนิภาพร ประภาศิริ
รหัสนักศึกษา	38626037
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2544
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร. เอื้อน ปิ่นเงิน

บทคัดย่อ

งานวิจัยนี้ได้ทำการศึกษาและวิเคราะห์การประยุกต์ตัววัดความซับซ้อนของซอฟต์แวร์ 3 ชนิด คือ ตัววัดของ Halstead (Halstead's Metric) วัดค่าความซับซ้อนจากขนาดของโปรแกรม ตัววัดไซโคลเมตริกของ McCabe (McCabe's Cyclomatic Metric) วัดค่าความซับซ้อนจากโครงสร้างการควบคุมการไหล (Control Flow Structure) และตัววัดของ Oviedo (Oviedo's Metric) วัดความซับซ้อนแบบผสม (Hybrid metric) ระหว่างโครงสร้างการควบคุมการไหลและโครงสร้างการไหลของข้อมูล (Data Flow Structure) ผู้วิจัยนำโปรแกรมประยุกต์ใช้ที่พัฒนาขึ้นด้วยภาษาวิซวลเบสิกจากบริษัทต่าง ๆ ในประเทศไทย เป็นเครื่องมือในการทดสอบตัววัดทั้งสาม พบว่าโปรแกรมที่มีโครงสร้าง Nested-If เมื่อเปลี่ยนเป็นโครงสร้าง Select-Case ที่ทำงานเหมือนกัน (Equivalent) มีตัววัดบางตัวให้ผลที่ไม่สอดคล้องกัน ผู้วิจัยจึงได้นำเสนอวิธีการวัดแบบใหม่โดยใช้โครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล ซึ่งเมื่อทดสอบกับโครงสร้าง Nested-If โครงสร้าง Select-Case และ โครงสร้างอื่น ๆ ของภาษาวิซวลเบสิกดังกล่าวแล้วให้ค่าเป็นที่น่าสนใจ

Thesis Title	Software Complexity Measurement
Student	Miss. Nipaporn Prapasiri
Student ID	38626037
Degree	Master of Science
Programme	Information Technology
Year	2001
Thesis Advisor	Asst. Prof. Dr. Ouen Pinngern

ABSTRACT

This research studies and analyzes the application of three software complexity metrics : Halstead's metric measures complexity of program size, McCabe's cyclomatic metric measures complexity of control flow structure, and Oviedo's hybrid metric measures complexity which combines control flow structure and data flow structure. We use Visual Basic application program code of companies in Thailand for testing of those metrics. When we test nested-If structure comparable to Select-Case structure that are equivalent, we observe some conflict results among the tested metrics. So we propose a new metric by using control flow structure and control flow data. When the nested-If structure Select-Case structure and other of Visual Basic program were tested, the results yield satisfactory.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำและคำปรึกษาเกี่ยวกับการดำเนินการวิจัยและแนวทางการแก้ปัญหาต่าง ๆ จาก ผศ.ดร. เอื้อน ปิ่นเงิน ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณเป็นอย่างสูง ขอขอบพระคุณบิดา-มารดา และพี่ ๆ ของข้าพเจ้า ซึ่งให้กำลังใจ และสนับสนุนทุนทรัพย์มาโดยตลอด

ขอขอบคุณ คุณวีรนุช วิไลพันธ์ คุณเอกชนก กฤตยาภาศิริวัฒน์ คุณพิชญ์ บุญล้อม คุณสิทธิศักดิ์ ทองสุข และคุณอลงกรณ์ พลชาติ ผู้ติดต่อประสานงานและอนุเคราะห์โปรแกรมภาษาวิซวลเบสิก เพื่อนำมาทำการทดสอบวัดค่าความซับซ้อนของโปรแกรมระบบงาน

ขอขอบคุณ คุณอดุลย์ ไตรมรรค ที่ให้กำลังใจและคำปรึกษา และห่วงใยในยามที่ข้าพเจ้ารู้สึกท้อ และเฝ้าดูความสำเร็จของข้าพเจ้าอย่างใกล้ชิดตลอดมา

ขอขอบคุณ คุณพัฒนพงษ์ ฉันทมิตรโอภาส และคุณวรางคณา เงินแก้ว คุณสรศักดิ์ ไทยแท้ และคุณปิยะวรรณ กระแสร์สุนทร ที่ให้กำลังใจซึ่งกันและกัน และให้คำปรึกษา พร้อมทั้งอำนวยความสะดวกและอุปการะต่าง ๆ ทำให้วิทยานิพนธ์นี้สำเร็จลุล่วงได้

ขอขอบคุณ เพื่อน ๆ วิทยาการสารสนเทศ รุ่น 1 และ พี่ ๆ ทุกคนในคณะเทคโนโลยีสารสนเทศ ที่ให้กำลังใจตลอดมา

สุดท้ายขอขอบคุณบัณฑิตวิทยาลัย ที่ได้ให้ทุนสนับสนุนการทำวิทยานิพนธ์ครั้งนี้ คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์นี้ ผู้วิจัยขอบอบแด่ผู้มีพระคุณทุกท่าน

นิภาพร ประภาศิริ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของการวิจัย	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 คำจำกัดความที่ใช้ในการวิจัย	2
1.4 ขอบเขตของการวิจัย	3
1.5 ลำดับการนำเสนอการวิจัย	3
บทที่ 2 ตัววัดความซับซ้อนของซอฟต์แวร์	4
2.1 ความซับซ้อนของซอฟต์แวร์	4
2.1.1 ตัววัดแบบไดนามิก	4
2.1.2 ตัววัดแบบสแตติก	4
2.2 ตัววัดความซับซ้อนของซอฟต์แวร์	5
2.2.1 ตัววัดขนาด	5
2.2.2 ตัววัดโครงสร้างข้อมูล	5
2.2.3 ตัววัดโครงสร้างตรรกะ	5
2.2.4 ตัววัดแบบผสม	5
2.3 งานวิจัยที่เกี่ยวกับตัววัดความซับซ้อน	6
2.3.1 ตัววัดจำนวนบรรทัด	6
2.3.2 ตัววัดของ Halstead	6
2.3.3 ตัววัดค่าไซโคลเมตริกของ McCabe	10
2.3.4 ตัววัดปม	13

สารบัญ (ต่อ)

	หน้า
2.3.5 ตัววัดขอบเขต	14
2.3.6 ตัววัดของ Chen	16
2.3.7 ตัววัดของ McClure	18
2.3.8 ตัววัดของ Henry และ Kafura	19
2.3.9 ตัววัดของ Shepperd และ Ince	21
2.3.10 ตัววัดของ Oviedo	21
บทที่ 3 ปัจจัยและการทดสอบวัดความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo	26
3.1 ปัจจัยตัววัดของ Halstead	26
3.2 ปัจจัยตัววัดของ McCabe	27
3.1 ปัจจัยตัววัดของ Oviedo	29
3.2 การทดสอบวัดความซับซ้อนของโปรแกรมจากตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo	30
บทที่ 4 การออกแบบตัววัดความซับซ้อนของซอฟต์แวร์จากโครงสร้างควบคุมการไหล และข้อมูลควบคุมการไหล	51
4.1 ความเป็นมาเกี่ยวกับตัววัด โครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล	51
4.2 ความซับซ้อนจากโครงสร้างควบคุมการไหล	52
4.3 ความซับซ้อนจากข้อมูลควบคุมการไหล	53
4.4 ตัววัดโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล	53
4.5 ปัจจัยของตัววัด SDM	54
4.6 การทดสอบวัดค่าความซับซ้อนจากตัววัด SDM	56
4.7 การเปรียบเทียบตัววัด SDM กับตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo	59

สารบัญ (ต่อ)

	หน้า
บทที่ 5 ผลการทดสอบการวัดความซับซ้อนของซอฟต์แวร์	64
5.1 ข้อมูลที่ใช้ในการทดสอบ	64
5.2 ผลการทดสอบการวัดค่าความซับซ้อน	64
5.3 เกณฑ์ความซับซ้อนที่เหมาะสมของโปรแกรม	65
5.4 ความสัมพันธ์ของความซับซ้อนกับจำนวนบรรทัดของโปรแกรม	67
บทที่ 6 บทสรุปและข้อเสนอแนะ	71
6.1 บทสรุป	71
6.2 ข้อเสนอแนะ	74
6.3 แนวทางการวิจัยต่อ	75
เอกสารอ้างอิง	76
ภาคผนวก	78
ภาคผนวก ก. คำสงวนภาษาวิซวลเบสิก	79
ภาคผนวก ข. ฟังก์ชันภาษาวิซวลเบสิก	82
ภาคผนวก ค. ตารางผลการทดสอบวัดความซับซ้อนของซอฟต์แวร์	84
ภาคผนวก ง. หน้าจอแสดงผลความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM	113
ภาคผนวก จ. บทความวิชาการและผลงานที่ได้รับการตีพิมพ์	116
ประวัติผู้เขียน	154

สารบัญตาราง

ตารางที่	หน้า
2.1 โอเปอเรเตอร์พื้นฐานในภาษาวิซวลเบสิก	7
2.2 ผลการนับค่า n_1 , n_2 , N_1 และ N_2	8
2.3 การหาค่าความซับซ้อนจากตัววัดขอบเขต	15
2.4 ผลสรุปค่าความซับซ้อนโดยตัววัดของ McClure	19
2.5 ค่าความซับซ้อนการไหลของข้อมูล	23
3.1 ประเภทและสัญลักษณ์ที่ใช้แทนประเภทของตัวแปร	26
3.2 จำนวน โอเปอเรเตอร์และโอเปอเรนด์ของโปรแกรมทดสอบที่ 1	31
3.3 ตัวแปรดีแฟกเฉพาะถิ่น การกำหนดค่าตัวแปรครั้งก่อน และความซับซ้อนจากการไหล.. ของข้อมูล ของโปรแกรมทดสอบที่ 1	32
3.4 ค่าความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และ ตัววัดของ Oviedo ของโปรแกรมทดสอบที่ 1	32
3.5 จำนวน โอเปอเรเตอร์และโอเปอเรนด์ของโปรแกรมทดสอบที่ 2	34
3.6 ตัวแปรดีแฟกเฉพาะถิ่น การกำหนดค่าตัวแปรครั้งก่อน และความซับซ้อนจากการไหล.. ของข้อมูล ของโปรแกรมทดสอบที่ 2	34
3.7 ค่าความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และ ตัววัดของ Oviedo ของโปรแกรมทดสอบที่ 2	35
3.8 จำนวน โอเปอเรเตอร์และโอเปอเรนด์ของโปรแกรมทดสอบที่ 3	37
3.9 ตัวแปรดีแฟกเฉพาะถิ่น การกำหนดค่าตัวแปรครั้งก่อน และความซับซ้อนจากการไหล .. ของข้อมูล ของโปรแกรมทดสอบที่ 3	37
3.10 ค่าความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และ ตัววัดของ Oviedo ของโปรแกรมทดสอบที่ 3	38
3.11 จำนวน โอเปอเรเตอร์และโอเปอเรนด์ของโปรแกรมทดสอบที่ 4	40
3.12 ตัวแปรดีแฟกเฉพาะถิ่น การกำหนดค่าตัวแปรครั้งก่อน และความซับซ้อนจากการไหล.. ของข้อมูล ของโปรแกรมทดสอบที่ 4	41
3.13 ค่าความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และ ตัววัดของ Oviedo ของโปรแกรมทดสอบที่ 4	42
3.14 จำนวน โอเปอเรเตอร์และโอเปอเรนด์ของโปรแกรมทดสอบที่ 5	44

สารบัญตาราง(ต่อ)

ตารางที่	หน้า
3.15 ค่าความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และ ตัววัดของ Oviedo ของโปรแกรมทดสอบที่ 5	45
3.16 จำนวนโอเปอเรเตอร์และโอเปอเรนด์ของโปรแกรมทดสอบที่ 6	47
3.17 ตัวแปรตีแผ่เฉพาะถิ่น การกำหนดค่าตัวแปรครั้งก่อน และความซับซ้อนจากการไหล.. ของข้อมูล ของโปรแกรมทดสอบที่ 6	48
3.18 ค่าความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และ ตัววัดของ Oviedo ของโปรแกรมทดสอบที่ 6	49
4.1 จำนวนข้อมูลควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมทดสอบที่ 1	56
4.2 จำนวนข้อมูลควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมทดสอบที่ 2	57
4.3 จำนวนข้อมูลควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมทดสอบที่ 3	57
4.4 จำนวนข้อมูลควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมทดสอบที่ 4	58
4.5 จำนวนข้อมูลควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมทดสอบที่ 5	58
4.6 จำนวนข้อมูลควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมทดสอบที่ 6.....	59
4.7 ผลการวัดความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM ของโปรแกรมทดสอบที่ 1 กับโปรแกรมทดสอบที่ 2	59
4.8 ผลการวัดความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM ของโปรแกรมรูปที่ 4.4 (ก) กับ โปรแกรมรูปที่ 4.4 (ข).....	61
4.9 ผลการวัดความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM ของโปรแกรมรูปที่ 4.4 (ค) กับ โปรแกรมรูปที่ 4.4 (ง)	62
5.1 จำนวนไฟล์และโพธิ์เซอร์ของ โปรแกรมทดสอบ	64

สารบัญรูป

รูปที่	หน้า
1.1 ความสัมพันธ์ของความซับซ้อนของซอฟต์แวร์	1
2.1 การแบ่งประเภทของตัววัดซอฟต์แวร์	4
2.2 ตัวอย่างโปรแกรมภาษาวิซวลเบสิก	8
2.3 สรุปผลการวัดความซับซ้อนของ Halstead	9
2.4 โปรแกรมและกราฟควบคุมการไหลของโปรแกรม	10
2.5 กราฟควบคุมการไหลเฉพาะ	11
2.6 การหาค่าความซับซ้อนจากกราฟควบคุมการไหล	12
2.7 ปมจากกราฟควบคุมการไหล	13
2.8 การควบคุมการไหลจากบรรทัดการทำงานของโปรแกรม	14
2.9 กราฟควบคุมการไหลของตัววัดขอบเขต	15
2.10 การหาค่า MIN จากกราฟควบคุมการไหล	16
2.11 กราฟควบคุมการไหลตัวอย่างการวัดของ Chen	17
2.12 โปรแกรมตัวอย่างการวัดของ McClure	19
2.13 การคำนวณการวัดความซับซ้อนในการไหลของข้อมูลของ Henry และ Kafura	20
2.14 โปรแกรมและกราฟควบคุมการไหลของโปรแกรม	23
3.1 ก ขอบเขตของโหนดและกราฟควบคุมการไหลของโครงสร้างโปรแกรม If ... Then	28
3.1 ข ขอบเขตของโหนดและกราฟควบคุมการไหลของโครงสร้างโปรแกรม If ... Else	28
3.1 ค ขอบเขตของโหนดและกราฟควบคุมการไหลของโครงสร้างโปรแกรม Do While	28
3.2 ก โปรแกรมทดสอบที่ 1	30
3.2 ข กราฟควบคุมการไหลของโปรแกรมทดสอบที่ 1	30
3.3 ก โปรแกรมทดสอบที่ 2	33
3.3 ข กราฟควบคุมการไหลของโปรแกรมทดสอบที่ 2	33
3.4 ก โปรแกรมทดสอบที่ 3	36
3.4 ข กราฟควบคุมการไหลของโปรแกรมทดสอบที่ 3	36
3.5 ก โปรแกรมทดสอบที่ 4	39
3.5 ข กราฟควบคุมการไหลของโปรแกรมทดสอบที่ 4	39
3.6 ก โปรแกรมทดสอบที่ 5	43

สารบัญรูป(ต่อ)

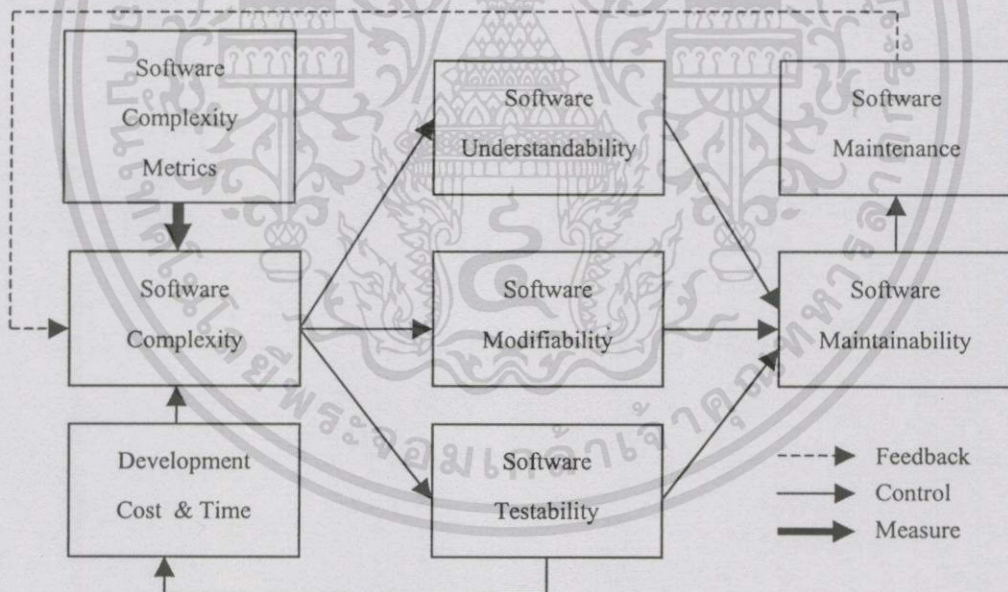
รูปที่	หน้า
3.6 ข กราฟควบคุมการไหลของโปรแกรมทดสอบที่ 5	43
3.7 ก โปรแกรมทดสอบที่ 6	46
3.7 ข กราฟควบคุมการไหลของโปรแกรมทดสอบที่ 6	46
4.1 องค์ประกอบหลักของซอฟต์แวร์	51
4.2 ก โครงสร้างโปรแกรมที่ประกอบด้วยกราฟควบคุมการไหลเฉพาะ D_1	52
4.2 ข โครงสร้างโปรแกรมที่ประกอบด้วยกราฟควบคุมการไหลเฉพาะ C_4	52
4.2 ค โครงสร้างโปรแกรมที่ประกอบด้วยกราฟควบคุมการไหลเฉพาะ D_0 ซึ่งซ้อนในกัน ...	52
4.2 ง โครงสร้างโปรแกรมที่ประกอบด้วยกราฟควบคุมการไหลเฉพาะ D_1 เรียงลำดับ D_0	52
4.3 ก กราฟควบคุมการไหลเฉพาะ P	55
4.3 ข กราฟควบคุมการไหลเฉพาะ D_0	55
4.3 ค กราฟควบคุมการไหลเฉพาะ D_1	55
4.3 ง กราฟควบคุมการไหลเฉพาะ D_2	55
4.3 จ กราฟควบคุมการไหลเฉพาะ D_3	55
4.3 ฉ กราฟควบคุมการไหลเฉพาะ D_4	55
4.3 ช กราฟควบคุมการไหลเฉพาะ C	55
4.4 โปรแกรมเปรียบเทียบความซับซ้อนจากตัววัด SDM กับตัววัดอื่น	60
5.1 กราฟแสดงค่าความซับซ้อนจากตัววัดของ Halstead เปรียบเทียบกับตัววัดของ McCabe	66
5.2 กราฟแสดงค่าความซับซ้อนจากตัววัดของ Oviedo เปรียบเทียบกับตัววัดของ McCabe ..	66
5.3 กราฟแสดงค่าความซับซ้อนจากตัววัด SDM เปรียบเทียบกับตัววัดของ McCabe	67
5.4 กราฟแสดงความสัมพันธ์ของความซับซ้อนจากตัววัดของ Halstead กับจำนวนบรรทัด ของโปรแกรม	68
5.5 กราฟแสดงความสัมพันธ์ของความซับซ้อนจากตัววัดของ McCabe กับจำนวนบรรทัด ของโปรแกรม	69
5.6 กราฟแสดงความสัมพันธ์ของความซับซ้อนจากตัววัดของ Oviedo กับจำนวนบรรทัด ของโปรแกรม	69
5.7 กราฟแสดงความสัมพันธ์ของความซับซ้อนจากตัววัด SDM กับจำนวนบรรทัดของ... ของโปรแกรม	70

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของการวิจัย

ปัจจุบันคุณภาพของซอฟต์แวร์มีบทบาทสำคัญต่อการพัฒนาซอฟต์แวร์และการบำรุงรักษาซอฟต์แวร์อย่างมาก [1] ความซับซ้อนของซอฟต์แวร์ถือเป็นคุณสมบัติหนึ่งของซอฟต์แวร์ที่สามารถนำไปใช้ในการประเมินคุณภาพของซอฟต์แวร์ได้ หากโปรแกรมที่เขียนขึ้นขาดคุณสมบัติที่ดี มีความซับซ้อนมาก มีโครงสร้างโปรแกรมไม่ดี ก็จะส่งผลต่อการบำรุงรักษา การทดสอบ และการแก้ไขเปลี่ยนแปลงโปรแกรมก็จะยุ่งยากมากขึ้นด้วย ความสัมพันธ์ของความซับซ้อนของซอฟต์แวร์กับกิจกรรมที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์แสดงได้ดังรูปที่ 1.1 [1] ด้วยความสำคัญดังกล่าวผู้วิจัยจึงได้ทำการศึกษาวิธีการวัดความซับซ้อนของซอฟต์แวร์แบบต่าง ๆ เพื่อพิจารณาข้อจำกัดของแต่ละวิธี แล้วหาวิธีการที่เหมาะสม เพื่อนำมาใช้ในการวัดความซับซ้อนของซอฟต์แวร์ได้อย่างมีประสิทธิภาพ



รูปที่ 1.1 ความสัมพันธ์ของความซับซ้อนของซอฟต์แวร์

นักวิชาการที่สร้างทฤษฎีหาค่าความซับซ้อนแต่ละแบบต่างมีมุมมองที่แตกต่างกันและตัววัดแต่ละแบบก็เหมาะสมกับลักษณะงานที่ต่างกัน ดังนั้นการใช้ตัววัดเพียงตัวเดียวสำหรับระบบงานหนึ่ง ๆ อาจไม่เหมาะสมเพราะไม่สามารถระบุถึงสิ่งที่ต้องการทราบและต้องการควบคุมได้ทั้งหมด

ตัววัดความซับซ้อนของซอฟต์แวร์ที่ได้มีการพัฒนามาสามารถแบ่งได้เป็น 2 ประเภท คือ แบบสแตติกและแบบไดนามิก [1] ตัววัดแบบสแตติกจะทำการวัดค่าความซับซ้อนโดยวิเคราะห์จากรหัสต้นฉบับ (Source Code) ส่วนแบบไดนามิกจะวัดในช่วงเวลาที่โปรแกรมนั้นทำงาน สำหรับการวิจัยนี้จะทำการศึกษาเฉพาะแบบสแตติกเท่านั้น

ตัววัดที่ทำการศึกษาในการวิจัยนี้ประกอบด้วย ตัววัดของ Halstead (Halstead's Metric) ซึ่งวัดค่าความซับซ้อนจากขนาดของโปรแกรม ตัววัดไซโคลเมตริกของ McCabe (McCabe Cyclomatic Metric) ซึ่งวัดค่าความซับซ้อนจากโครงสร้างการควบคุมการไหล (Control Flow Structure) และตัววัดของ Oviedo (Oviedo's Metric) ซึ่งวัดความซับซ้อนแบบผสมระหว่างโครงสร้างการควบคุมการไหลและโครงสร้างการไหลของข้อมูล (Data Flow Structure) และได้ทำการวิเคราะห์ผลการวัดความซับซ้อนของโปรแกรมซึ่งทำให้มีการเสนอวิธีการวัดแบบใหม่

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาวิธีการวัดความซับซ้อนของซอฟต์แวร์แบบสแตติกที่มีอยู่ ว่าแต่ละวิธีมีข้อจำกัดอย่างไร
2. เพื่อนำเสนอตัววัดความซับซ้อนที่เหมาะสม โดยพิจารณาว่ามีคุณสมบัติใดบ้างที่ส่งผลต่อความซับซ้อนของโปรแกรม
3. เพื่อนำเสนอเกณฑ์การวัดความซับซ้อนของโปรแกรมที่เหมาะสมสำหรับการพัฒนาซอฟต์แวร์ โดยแต่ละโพรซีเจอร์ควรมีค่าความซับซ้อนเท่าใด

1.3 คำจำกัดความที่ใช้ในการวิจัย

ตัววัดซอฟต์แวร์ (Software Metrics) หมายถึง วิธีมาตรฐานที่ใช้ในการวัดคุณสมบัติของซอฟต์แวร์ เช่น ขนาด ค่าใช้จ่าย ความซับซ้อนและความน่าเชื่อถือของโปรแกรม [2]

ความซับซ้อนของซอฟต์แวร์ (Software Complexity) หมายถึง ความยากต่อการเข้าใจ การแก้ไขเปลี่ยนแปลง และการทดสอบโปรแกรม [3]

จำนวนบรรทัด (Line of code) หมายถึง จำนวนบรรทัดของข้อความในโปรแกรม (Program Text) ซึ่งไม่รวมบรรทัดหมายเหตุ (Comment) และบรรทัดว่าง (Blank Line) ที่ปรากฏในโปรแกรม

1.4 ขอบเขตของการวิจัย

1. งานวิจัยนี้ศึกษาตัววัดความซับซ้อนเฉพาะตัววัดแบบสแตติก
2. งานวิจัยนี้เลือกศึกษาตัววัดความซับซ้อนของซอฟต์แวร์ต่อไปนี้
 - 1.1 ตัววัดของ Halstead
 - 1.2 ตัววัดไซโคลเมตริกของ McCabe
 - 1.3 ตัววัดของ Oviedo
3. โปรแกรมที่จะนำมาวัดความซับซ้อนต้องเป็นโปรแกรมที่พัฒนาด้วยภาษาวิชวลเบสิก ไม่เกินเวอร์ชัน 5 เป็นการวัดความซับซ้อนโดยพิจารณาหัสต์ต้นฉบับที่เป็น Procedural
4. โปรแกรมที่นำมาวัดต้องเป็นโปรแกรมที่มีความถูกต้องทางไวยากรณ์ของภาษาวิชวลเบสิกแล้ว

1.5 ลำดับการนำเสนอการวิจัย

ในการวิจัยนี้ได้นำเสนอวิธีการวัดความซับซ้อนของซอฟต์แวร์ โดยรายละเอียดของการวิจัยและเนื้อหาแบ่งออกได้ดังนี้

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของการวิจัย คำจำกัดความในการวิจัยและขอบเขตการวิจัย

บทที่ 2 ตัววัดความซับซ้อนของซอฟต์แวร์ ในบทนี้จะกล่าวถึงความหมายของความซับซ้อนของซอฟต์แวร์ ประเภทของตัววัดความซับซ้อนซอฟต์แวร์ และงานวิจัยที่เกี่ยวกับตัววัดความซับซ้อน

บทที่ 3 ปัจจัยและการทดสอบวัดความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo ในบทนี้นำเอาตัววัดในบทที่ 2 ได้แก่ ตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo มาพิจารณาปัจจัยตามโครงสร้างภาษาวิชวลเบสิก และทดสอบวัดค่าความซับซ้อนด้วยตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo

บทที่ 4 การออกแบบตัววัด โครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล เป็นการนำเสนอตัววัดความซับซ้อนใหม่ รวมถึงการพิจารณาปัจจัยและทำการทดสอบตัววัด พร้อมทั้งเปรียบเทียบตัววัดแบบใหม่นี้กับตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo

บทที่ 5 แสดงผลการทดสอบการวัดความซับซ้อนของซอฟต์แวร์และนำเสนอเกณฑ์การวัดความซับซ้อนที่เหมาะสม และแสดงความสัมพันธ์ของความซับซ้อนที่ได้จากตัววัดต่าง ๆ กับจำนวนบรรทัดของโปรแกรม

บทที่ 6 บทสรุปและข้อเสนอแนะ กล่าวถึงผลสรุปของการวิจัย ข้อเสนอแนะและแนวทางการวิจัยต่อไป

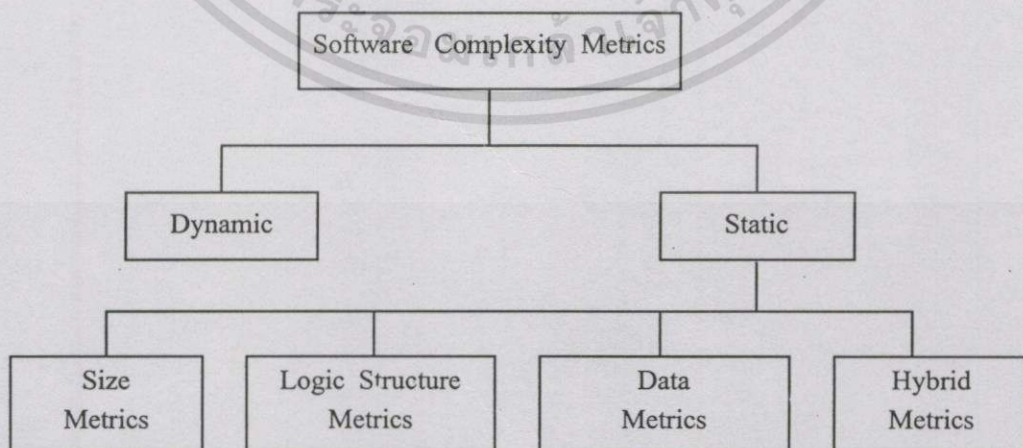
ตัววัดความซับซ้อนของซอฟต์แวร์

2.1 ความซับซ้อนของซอฟต์แวร์

ความซับซ้อนของซอฟต์แวร์ (Software Complexity) หมายถึง ความยากต่อการเข้าใจ การแก้ไขเปลี่ยนแปลง และการทดสอบโปรแกรม [3] ซึ่งซอฟต์แวร์ที่มีความซับซ้อนมากจะส่งผลกระทบต่อต้นทุนและเวลาในการพัฒนาซอฟต์แวร์ด้วย ซึ่งถือเป็นปัจจัยที่สำคัญที่สุด ด้วยเหตุนี้นักวิจัยจึงได้พยายามสร้างตัววัดความซับซ้อน เพื่อให้ค่าความซับซ้อนที่วัดได้เป็นไปตามความเป็นจริงมากที่สุด จากรูปที่ 2.1 แสดงการแบ่งประเภทตัววัดความซับซ้อนของซอฟต์แวร์ที่มีการพัฒนากันมาแบ่งได้เป็น 2 ประเภท [1] ดังนี้

2.1.1 ตัววัดแบบไดนามิก (Dynamic) ทำการวัดความซับซ้อนในช่วงเวลาที่โปรแกรมทำงาน ซึ่งปัจจัยที่ใช้เป็นคุณสมบัติที่วัดได้โดยพิจารณาขณะที่โปรแกรมทำงาน เช่น การวัดค่า Big-O จะพิจารณาช่วงเวลาที่ใช้ในการทำงานของโปรแกรม

2.1.2 ตัววัดแบบสถิติก (Static) ทำการวัดความซับซ้อนโดยวิเคราะห์จากรหัสต้นฉบับ (Source Code) ซึ่งปัจจัยที่ใช้เป็นคุณสมบัติที่สามารถวัดได้โดยตรงจากรหัสต้นฉบับ เช่น จำนวนบรรทัดของโปรแกรม จำนวนตัวแปร จำนวนโครงสร้างตรรกะ เป็นต้น และจากตัววัดที่มีการวิจัยกันมาแบ่งได้ 4 ลักษณะ คือ ตัววัดขนาด (Size Metrics) ตัววัดโครงสร้างตรรกะ (Logic Structure Metrics) ตัววัดข้อมูล (Data Metrics) และตัววัดแบบผสม (Hybrid Metrics) ซึ่งจะได้กล่าวถึงรายละเอียดในหัวข้อต่อไป



รูปที่ 2.1 การแบ่งประเภทของตัววัดซอฟต์แวร์

2.2 ตัววัดความซับซ้อนของซอฟต์แวร์

งานวิจัยนี้ผู้วิจัยศึกษาเฉพาะการวัดแบบสแตติกเท่านั้นซึ่งจะพิจารณาคูณสมบัติของความซับซ้อนได้จากรหัสต้นฉบับโดยตรง ตัววัดประเภทนี้แบ่งออกเป็น 4 ลักษณะ [2] ดังนี้

2.2.1 ตัววัดขนาด (Size Metrics) เป็นวิธีวัดที่ง่าย และนิยมใช้กันมาก ตัววัดที่จัดอยู่ในประเภทนี้ ได้แก่ ตัววัดจำนวนบรรทัดของโปรแกรม (Line of code) ซึ่งเชื่อว่าโปรแกรมที่มีจำนวนบรรทัดมากน่าที่จะมีความซับซ้อนมากกว่าโปรแกรมที่มีจำนวนบรรทัดน้อยกว่า และตัววัดของ Halstead (Halstead's Metric) ซึ่งจะพิจารณาจากโอเปอเรเตอร์ และโอเปอเรนด์ นำไปใช้ในการวัดความยาวของโปรแกรม การวัดปริมาตร และการวัดระดับของโปรแกรม ต่อมา Halstead ได้กำหนดสูตรเพื่อใช้ในการวัดความพยายามของโปรแกรม ซึ่งสามารถแสดงถึงความซับซ้อนได้ กล่าวคือโปรแกรมใดที่มีค่าความพยายามมาก โปรแกรมนั้นจะมีความซับซ้อนมากด้วย

2.2.2 ตัววัดโครงสร้างตรรกะ (Logic Structure Metrics) หรือตัววัดโครงสร้างการควบคุมการไหล (Control Flow Structure Metric) เป็นการหาค่าความซับซ้อนจากกราฟของโปรแกรม เรียกกราฟนี้ว่า กราฟควบคุมการไหล (Control Flow Graph) ตัววัดที่จัดอยู่ในประเภทนี้ ได้แก่ ตัววัดค่าไซโคลเมตริกของ McCabe (McCabe's Cyclomatic Metric) ตัววัดปม (Knot Metric) และตัวขอบเขต (Scope Metric) ตัววัดของ Chen (Chen's Metric) ตัววัดของ McClure (McClure's Metric) เป็นต้น ตัววัดในกลุ่มนี้ส่วนใหญ่มีแนวความคิดเพื่อที่จะแก้ไขข้อจำกัดตัววัดของ McCabe ซึ่งโปรแกรมที่มีความซับซ้อนมากจะให้ค่าความซับซ้อนมากกว่าโปรแกรมที่มีความซับซ้อนน้อยกว่า

2.2.3 ตัววัดข้อมูล (Data Metrics) วิธีนี้คำนวณค่าความซับซ้อนโดยพิจารณาจาก การสื่อสารและการใช้ข้อมูลร่วมกันระหว่างโมดูล การจัดการและการกระจายของข้อมูล ค่าคงที่ และตัวแปรต่าง ๆ ภายในโปรแกรม ตัววัดที่จัดอยู่ในประเภทนี้ ได้แก่ ตัววัดของ Henry และ Kafura (Henry and Kafura's Metric) พิจารณาความซับซ้อนจากการสื่อสาร การใช้ข้อมูลร่วมกันระหว่างโมดูล ความยาวของโปรแกรม ส่วนตัววัดของ Shepperd และ Ince (Shepperd and Ince's Metric) ประยุกต์จากตัววัดของ Henry และ Kafura ซึ่งจะไม่พิจารณาความยาวของโปรแกรม เพราะเขาคิดว่าความซับซ้อนของโปรแกรมไม่ได้ขึ้นอยู่กับความยาวของโปรแกรม

2.2.4 ตัววัดแบบผสม (Hybrid Metrics) เป็นตัววัดค่าความซับซ้อนจากการไหลของข้อมูล (Data Flow) และจากกราฟควบคุมการไหล ตัววัดที่จัดอยู่ในประเภทนี้ ได้แก่ ตัววัดของ Oviedo (Oviedo's Metric)

จากตัววัดทั้ง 4 แบบ ตัววัดโครงสร้างตรรกะถือว่าเป็นตัววัดที่ได้รับความสนใจจากนักวิจัยมากที่สุด เนื่องจากนักวิจัยส่วนใหญ่เชื่อว่าความซับซ้อนของโปรแกรมน่าจะเกิดการโครงสร้างการเขียนโปรแกรมมากกว่า

2.3 งานวิจัยที่เกี่ยวกับตัววัดความซับซ้อน

ในปี ค.ศ. 1972 Halstead เป็นผู้ที่ทำให้เกิดสาขาย่อยของวิทยาการคอมพิวเตอร์ที่ชื่อว่า ตัววัดความซับซ้อนของซอฟต์แวร์ [4] ต่อมามีนักวิจัยหลายท่านพยายามหาวิธีการพิสูจน์และแก้ไขข้อจำกัดจากตัววัดของ Halstead นักวิจัยแต่ละท่านใช้ปัจจัยที่แตกต่างกันไม่ว่าจะเป็นการวัดจากโครงสร้างหรือการวัดจากขนาดของโปรแกรม เกิดจากการสังเกตเห็นว่าปัจจัยนั้นมีผลต่อความซับซ้อนของโปรแกรมแล้วนำปัจจัยเหล่านั้นมาหาความสัมพันธ์เพื่อนำเสนอแนวทางและวิธีการใหม่ซึ่งเชื่อว่าจะสามารถหาค่าความซับซ้อนของซอฟต์แวร์ได้ดียิ่งขึ้นและตรงตามความรู้สึกที่เป็นจริง ในส่วนนี้จะกล่าวถึงงานวิจัยที่เกี่ยวกับตัววัดความซับซ้อนที่น่าสนใจดังต่อไปนี้

2.3.1 ตัววัดจำนวนบรรทัด (Line of code) เป็นการวัดโดยนับจำนวนบรรทัดของข้อความในโปรแกรม (Program Text) ซึ่งไม่รวมบรรทัดหมายเหตุ (Comment Line) และบรรทัดว่าง (Blank Line) ที่ปรากฏในโปรแกรม [4] ซึ่งเชื่อกันว่าโปรแกรมที่ยาวกว่าน่าจะมีค่าความซับซ้อนมากกว่าโปรแกรมที่สั้นกว่า ในช่วงแรกมีการกำหนดการนับที่แตกต่างกันไป บางคนนับแต่คำสั่งที่มีการกระทำ (Execute) เท่านั้น บางคนก็นับทั้งคำสั่งที่มีการกระทำและคำสั่งที่เป็นการกำหนดข้อมูล หรือบางคนนับทุกคำสั่งที่อยู่ในตัวโปรแกรมรวมถึงประโยคหมายเหตุ (Comment) ด้วย [5] ซึ่งนักวิจัยส่วนใหญ่เห็นด้วยกับการนับเฉพาะคำสั่งที่มีการกระทำเท่านั้น

วิธีการนี้เป็นวิธีที่ได้รับความนิยมมาก เนื่องจากเป็นวิธีที่ง่ายในการวัดและเขียนโปรแกรมทำการวัด

2.3.2 ตัววัดของ Halstead (Halstead's Metric) Halstead เป็นผู้ให้กำเนิดวิทยาการการวัดความซับซ้อนของซอฟต์แวร์ โดยพิจารณาจากโอเปอเรเตอร์และโอเปอแรนด์ ที่มีอยู่ในโปรแกรม ซึ่งกำหนดไว้ดังนี้

โอเปอเรเตอร์ แบ่งเป็น 3 กลุ่ม [6] ได้แก่

1. โอเปอเรเตอร์พื้นฐาน ดังแสดงในตารางที่ 2.1
2. โอเปอเรเตอร์ที่เป็นคำสงวน (Reserved Words) แสดงไว้ในภาคผนวก ก.
3. โอเปอเรเตอร์พิเศษ ได้แก่ ชื่อตัวแปร และชื่อฟังก์ชันแสดงไว้ในภาคผนวก ข.

โอเปอแรนด์ หมายถึง ตัวแปร และค่าคงที่ที่ใช้ในโปรแกรม

ตารางที่ 2.1 โอเปอเรเตอร์พื้นฐานในภาษาวิซวลเบสิก

กลุ่มของโอเปอเรเตอร์	สัญลักษณ์	ความหมาย
คณิตศาสตร์	+ - * / ^ \ Mod	บวก ลบ หรือ ค่าลบ คูณ หาร ยกกำลัง หารจำนวนเต็ม มอดคูลุโ (เศษ)
เปรียบเทียบ	= <> > < >= <=	เท่ากัน ไม่เท่ากัน มากกว่า น้อยกว่า มากกว่าหรือเท่ากับ น้อยกว่าหรือเท่ากับ
ตรรกศาสตร์	Not And Or Xor Eqv Imp	ไม่ และ หรือ เป็น จริง เมื่อโอเปอเรนด์ทั้งสองตัวต่างกัน นอก นั้นเป็น เท็จ เป็น จริง เมื่อโอเปอเรนด์ทั้งสองตัวเหมือนกัน นอกนั้นเป็น เท็จ เป็น เท็จ เมื่อโอเปอเรนด์ตัวแรกเป็น จริง และตัว ที่สองเป็น เท็จ นอกนั้นเป็น จริง
สตริง	&	เชื่อมต่อสตริง
อื่น ๆ	() [] , " .	วงเล็บ ปีกกา จุดภาค อัญประกาศ จุด

วิธีการของ Halstead มีตัวแปรที่เกี่ยวข้องดังนี้ [3]

n_1 คือ จำนวนโอเปอเรเตอร์ทั้งหมดที่ไม่ซ้ำกันในโปรแกรม (Unique Operators)

n_2 คือ จำนวนโอเปอเรนด์ทั้งหมดที่ไม่ซ้ำกันในโปรแกรม (Unique Operands)

N_1 คือ จำนวนโอเปอเรเตอร์ทั้งหมดในโปรแกรม (Total Operators)

N_2 คือ จำนวนโอเปอเรนด์ทั้งหมดในโปรแกรม (Total Operands)

จากโปรแกรมรูปที่ 2.2 สามารถแสดงค่า n_1 , n_2 , N_1 และ N_2 ได้ดังตารางที่ 2.2

```

Sub Text1_Change()
    a = Val(Text1)
    b = a Mod 2
    If (b = 0) Then
        Print a, "เป็นเลขจำนวนคู่"
    Else
        Print a, "เป็นเลขจำนวนคี่"
    End If
End Sub

```

รูปที่ 2.2 ตัวอย่างโปรแกรมภาษาวิซวลเบสิก

ตารางที่ 2.2 ผลการนับค่า n_1 , n_2 , N_1 และ N_2

โอเปอเรเตอร์	จำนวน	โอเปอเรนด์	จำนวน
Sub	1	.a	4
()	3	b	2
Val	1	2	1
=	3	0	1
Mod	1	Text1	1
If	1	"เป็นเลขจำนวนคู่"	1
Then	1	"เป็นเลขจำนวนคี่"	1
Print	2		
Else	1		
End If	1		
,	2		
""	2		
End Sub	1		
$n_1 = 13$	$N_1 = 20$	$n_2 = 7$	$N_2 = 11$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Halstead วัดความซับซ้อนโดยใช้ค่า n_1 n_2 N_1 และ N_2 ซึ่งสามารถพิจารณาความซับซ้อนได้จากสิ่งต่อไปนี้ [3] [7]

2.3.2.1 ความยาวของโปรแกรม (Program Length) Halstead เชื่อว่าโปรแกรมใดที่ยาวกว่าย่อมมีแนวโน้มที่จะมีความซับซ้อนมากกว่าโปรแกรมที่สั้นกว่า และความยาวนี้เป็นการนับจำนวนคำสั่ง (Statement) ของโปรแกรมที่มีการกระทำ (Execute) โดยกำหนดค่า N แทนความยาวของโปรแกรม คำนวณได้ดังนี้

$$N = N_1 + N_2 \quad (2.1)$$

หรือประมาณค่าได้จากสูตร

$$\hat{N} = n_1 \log_2 n_1 + n_2 \log_2 n_2 \quad (2.2)$$

2.2.3.2 ความพยายามของโปรแกรม (Program Effort) หมายถึง ค่าที่แสดงถึงความพยายามของจิตใจในการเขียนโปรแกรมนั้น ๆ หรือเป็นความพยายามที่ต้องใช้สำหรับอ่านและเข้าใจโปรแกรม Halstead เชื่อว่าโปรแกรมที่มีความซับซ้อนมาก น่าที่จะต้องใช้ความพยายามมากกว่าโปรแกรมที่มีความซับซ้อนน้อยกว่า

Halstead นิยามค่า E แทนความพยายามของโปรแกรมหาดังนี้

$$E = \frac{V}{L} \quad (2.3)$$

โดยที่ V แทนปริมาตรของโปรแกรม (Program Volume) เท่ากับ $N \log_2 (n_1 + n_2)$ และ L แทนระดับของโปรแกรม (Level of abstraction) ซึ่งเป็นปริมาณที่แสดงถึงระดับในทางนามธรรมของภาษาที่ใช้ในการเขียนโปรแกรมคำนวณได้จาก $\frac{2n_2}{n_1 N_2}$ ดังนั้นค่า E สามารถคำนวณได้จากสูตร

$$E = \frac{n_1 N_2 (N_1 + N_2) \log_2 (n_1 + n_2)}{2n_2} \quad (2.4)$$

จากค่า n_1 n_2 N_1 และ N_2 ในตารางที่ 2.2 นำมาวัดความซับซ้อนแบบต่าง ๆ ของ Halstead สามารถสรุปได้ดังรูปที่ 2.3 ตัววัดของ Halstead ทั้ง 2 แบบ พิจารณาจากโอเปอเรเตอร์และโอเปอเรนด์เท่านั้น ไม่มีการพิจารณาโครงสร้างควบคุมการไหล

ความยาวของโปรแกรม (N)	$20 + 11 = 31$
ความพยายามของโปรแกรม (E)	$13 \times 11 \times (20 + 11) \log_2(13 + 7) / 2 \times 7 = 1368.508$

รูปที่ 2.3 สรุปผลการวัดความซับซ้อนของ Halstead

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 ตัววัดไซโคลเมตริกของ McCabe (McCabe's Cyclomatic Metric)

วิธีการวัดของ McCabe [8] วัดจากกราฟควบคุมการไหล เป็นการจำลองโปรแกรมด้วยทิศทางของกราฟดังแสดงในรูปที่ 2.4 จากโปรแกรมรูปที่ 2.4 (ก) สามารถแสดงกราฟควบคุมการไหลของโปรแกรมได้ดังรูปที่ 2.4 (ข) โดยการแทนกลุ่มคำสั่งด้วยโหนด (Node) และเส้นเชื่อม (Edge) กราฟควบคุมการไหล [9] คือ กราฟที่มีทิศทางจากโหนดเริ่มต้น (Start Node) ถึงโหนดสิ้นสุด (Stop Node) โดยที่โหนดสิ้นสุดจะไม่มีทิศทางออกจากตัวเอง กราฟควบคุมการไหลเป็นโมเดลที่เหมาะสมสำหรับการวัดโครงสร้างการควบคุมการไหลของโปรแกรมเพราะแสดงให้เห็นโครงสร้างของโปรแกรมได้อย่างชัดเจน โดยแต่ละโหนดแทนกลุ่มคำสั่งแต่ละกลุ่มในโปรแกรม และเส้นเชื่อมแทนเส้นต่อระหว่างโหนดแต่ละโหนดซึ่งแสดงการควบคุมการไหลจากโหนดหนึ่งไปยังโหนดถัดไปบอกถึงลำดับก่อนหลังของการประมวลผลแต่ละโหนดการแยกคำสั่งต่าง ๆ ในโปรแกรมออกเป็นกลุ่มคำสั่งหรือโหนดทำได้โดยพิจารณา ถ้าคำสั่งแรกในกลุ่มถูกประมวลผลแล้ว คำสั่งทุก ๆ คำสั่งในกลุ่มนั้นต้องถูกประมวลผลด้วย [2] หรือพิจารณาคำสั่งประเภทการตัดสินใจ เช่น If While แทนหนึ่งโหนด และคำสั่งอื่น ๆ ที่มีการประมวลผลแบบลำดับแทนหนึ่งกลุ่มคำสั่งหรือหนึ่งโหนด

Sub Text1_Change()

a = Val(Text1)

b = a Mod 2

If (b = 0) Then

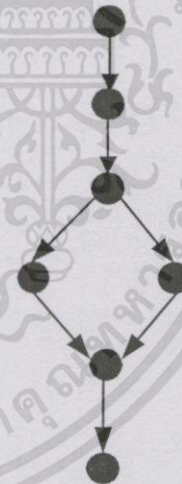
Print a, "เป็นเลขจำนวนคู่"

Else

Print a, "เป็นเลขจำนวนคี่"

End If

End Sub



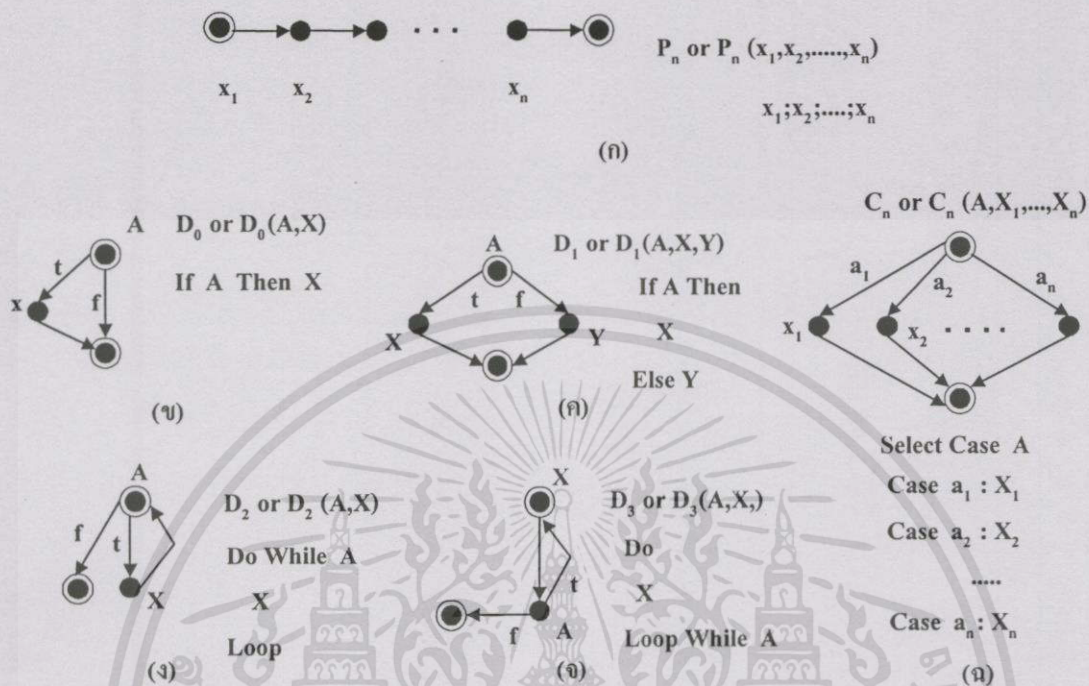
(ก) โปรแกรม

(ข) กราฟควบคุมการไหลของโปรแกรม

รูปที่ 2.4 ตัวอย่างโปรแกรมและกราฟควบคุมการไหลของโปรแกรมของตัววัด McCabe

จากคุณสมบัติการควบคุมการไหลของโปรแกรมจะเห็นว่าโปรแกรมหนึ่ง ๆ สามารถนำมาเขียนแสดงเป็นกราฟควบคุมการไหลได้เพียงรูปแบบเดียวเท่านั้น โดยกราฟควบคุมการไหลดังกล่าวประกอบด้วยโครงสร้างเฉพาะของกราฟควบคุมการไหลเฉพาะ (Prime Control Flow Graphs) หลาย ๆ กราฟประกอบกัน กราฟควบคุมการไหลเฉพาะ [9] คือ กราฟควบคุมการไหลที่ไม่

สามารถแตกย่อยไปเป็นการทำงานแบบลำดับ (Sequence) หรือการซ้อนใน (Nesting) ได้อีก เช่น กราฟควบคุมการไหลในรูปที่ 2.5 แสดงโครงสร้างเฉพาะของกราฟควบคุมการไหลแบบต่าง ๆ



(ก) กราฟควบคุมการไหล P_n (ข) กราฟควบคุมการไหล D_0
 (ค) กราฟควบคุมการไหล D_1 (ง) กราฟควบคุมการไหล D_2
 (จ) กราฟควบคุมการไหล D_3 (ฉ) กราฟควบคุมการไหล C_n

รูปที่ 2.5 กราฟควบคุมการไหลเฉพาะ

กราฟควบคุมการไหลรูปที่ 2.5 (ก) เรียกว่า P_n เป็นกราฟควบคุมการไหลแสดงการไหลของโปรแกรมที่มีการทำงานแบบลำดับจากโนดที่ x_1 ถึง โนดที่ x_n กราฟควบคุมการไหลรูปที่ 2.5 (ข) ถึงรูปที่ 2.5 (ฉ) แสดงโครงสร้างของกราฟควบคุมการไหลเฉพาะซึ่งเป็นโครงสร้างของกราฟควบคุมการไหลที่มีเงื่อนไขของโปรแกรมนั้น กราฟควบคุมการไหลรูปที่ 2.5 (ข) เงื่อนไขของโปรแกรมเป็น If A Then X เรียกว่ากราฟควบคุมการไหล D_0 กราฟควบคุมการไหลรูปที่ 2.5 (ค) เงื่อนไขของโปรแกรมเป็น If A Then X Else Y เรียกว่ากราฟควบคุมการไหล D_1 กราฟควบคุมการไหลรูปที่ 2.5 (ง) เงื่อนไขของโปรแกรมเป็น Do While A X เรียกว่ากราฟควบคุมการไหล D_2 กราฟควบคุมการไหลรูปที่ 2.5 (จ) เงื่อนไขของโปรแกรม เป็น Do X Loop A เรียกว่ากราฟควบคุมการไหล D_3 ส่วนรูปที่ 2.5 (ฉ) แสดงกราฟควบคุมการไหลที่มีเงื่อนไขเป็น Select Case เรียกกราฟควบคุมการไหลนี้ว่า C_n ซึ่งแสดง Case A จาก $a_1 : x_1$ ถึง $a_n : x_n$

McCabe กำหนดให้ F แทนกราฟควบคุมการไหลของโปรแกรม และกำหนดสูตรหาค่าความซับซ้อนจากค่าไซโคลเมตริกของกราฟควบคุมการไหลไว้ดังนี้

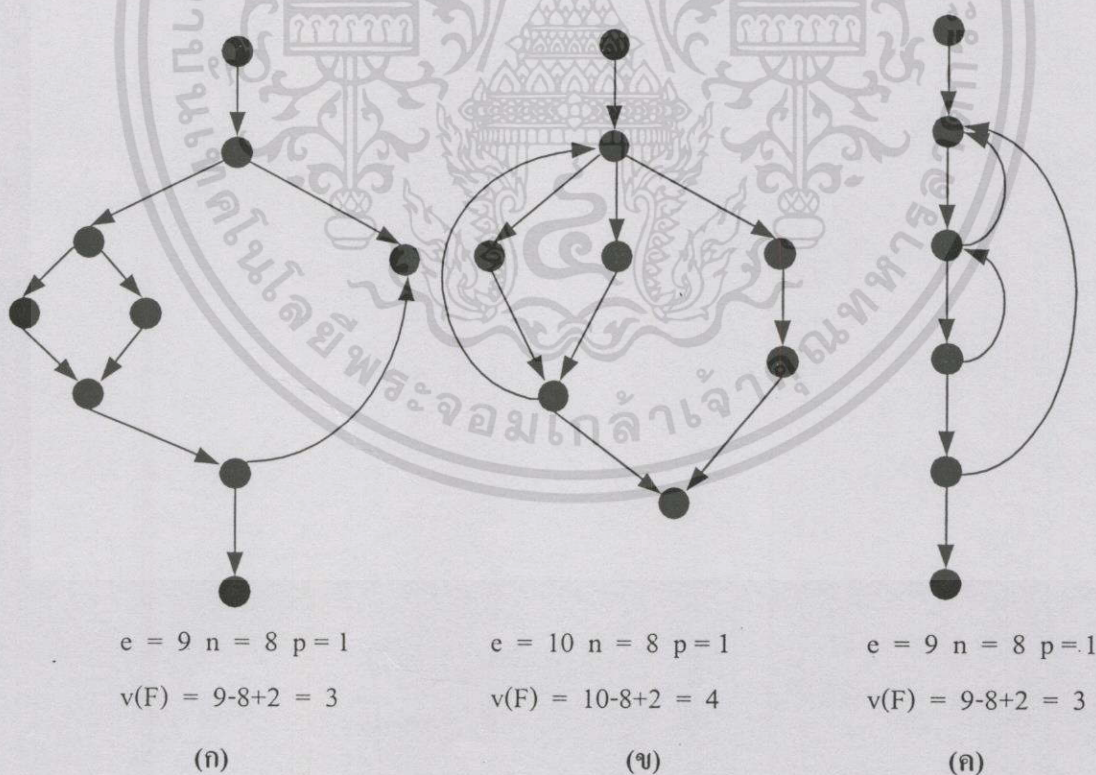
$$v(F) = e - n + 2p \tag{2.5}$$

โดยที่ e คือ จำนวนเส้นเชื่อมของกราฟ (Edges)

n คือ จำนวนโหนดของกราฟ (Nodes)

p คือ จำนวนคอมโพเนนต์หรือจำนวนโมดูล

จากรูปที่ 2.6 แสดงตัวอย่างหาค่าความซับซ้อนจากกราฟควบคุมการไหล จากตัวอย่างเห็นได้ชัดว่าค่า $v(F)$ ขึ้นกับโครงสร้างของการตัดสินใจของโปรแกรม ถ้าค่า $v(F)$ สูงโปรแกรมนั้นจะซับซ้อนมาก ดังนั้นการเขียนหรือออกแบบโปรแกรมจะต้องใช้เวลานานขึ้นส่งผลต่อค่าใช้จ่ายและการพัฒนาโปรแกรมจะสูงขึ้นด้วย เมื่อเปรียบเทียบการเขียนโปรแกรมของคนสองคนสามารถเขียนโปรแกรมได้ถูกต้องตามที่ต้องการทุกประการ แต่เมื่อนำทั้งสองโปรแกรมมาวัดค่า $v(F)$ ได้ผลว่าโปรแกรมแรกมีค่า $v(F)$ สูงกว่าโปรแกรมที่สองอาจกล่าวได้ว่าโปรแกรมที่สองควรจะดีกว่าโปรแกรมแรก



รูปที่ 2.6 การหาค่าความซับซ้อนจากกราฟควบคุมการไหล

McCabe ได้แนะนำวิธีการหาค่า $v(F)$ ไว้อีก 2 วิธีดังนี้

1. $v(F) = d+1$ โดยที่ d คือจำนวนโหนดที่มีทิศทางออกจากตัวเอง (Out Degree) มากกว่า 1
2. $v(F)$ เท่ากับจำนวนระนาบ (Region) ของกราฟควบคุมการไหล

McCabe แนะนำว่าแต่ละโมดูลไม่ควรมีค่า $v(F)$ เกิน 10 และโดยทั่วไปแล้วค่า $v(F)$ ที่เหมาะสมจะอยู่ระหว่าง 3 ถึง 7 และเขาพบว่าโปรแกรมที่มีค่า $v(F)$ น้อยจะสามารถทดสอบได้ง่ายและมีความน่าเชื่อถือสูง McCabe คิดว่าความยากในการเข้าใจโปรแกรมที่จริงแล้วขึ้นอยู่กับกราฟควบคุมการไหลของโปรแกรมนั้น ๆ

ตัววัดของ McCabe ไม่สะท้อนบอกถึงขนาดของโปรแกรมได้ โปรแกรมหนึ่งอาจใหญ่กว่าอีกโปรแกรมหนึ่งซึ่งทั้งสองโปรแกรมอาจมีค่า $v(F)$ เท่ากันได้ นอกจากนี้ตัววัดนี้ไม่สามารถบอกถึงความซับซ้อนในแง่การซ้อนใน (Nest) และการวนซ้ำ (Loop) ในโปรแกรม

2.3.4 ตัววัดปม (Knot Metric)

Woodward Hennell และ Hedley [10] ได้นำเสนอตัววัดปมของโปรแกรมนำมาใช้เป็นเกณฑ์แสดงถึงความซับซ้อน โดยพิจารณาจากโครงสร้างของโปรแกรม ซึ่งโปรแกรมจะเกิดปมก็ต่อเมื่อกราฟควบคุมการไหลสองเส้นทางตัดกันดังแสดงในรูปที่ 2.7 ถ้าโปรแกรมใดมีปมมากโปรแกรมนั้นก็ถือว่ามีความซับซ้อนมาก ตัววัดนี้จะพิจารณาจำนวนจุดตัดทั้งหมดของเส้นเชื่อมที่ควบคุมการไหล

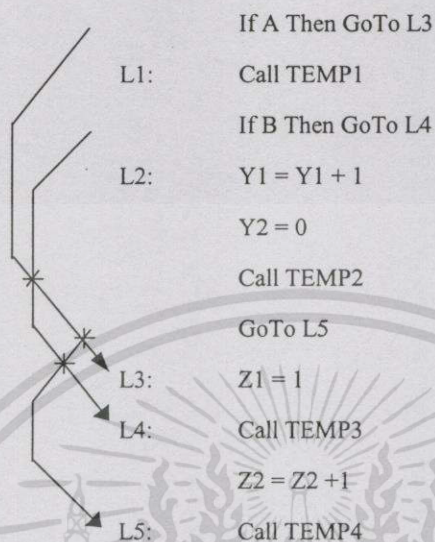


รูปที่ 2.7 ปมจากกราฟควบคุมการไหล

รูปแบบการวัดโปรแกรมควบคุมโครงสร้างนี้เป็นโปรแกรมที่มีลำดับและมีบรรทัดควบคุมลำดับการทำงานของโปรแกรมเพิ่มเข้ามาดังรูปที่ 2.8 แสดงการควบคุมการไหลจากบรรทัดการทำงานของโปรแกรมภาษาวิซวลเบสิก จะพบว่าจุดตัดที่เกิดขึ้นของบรรทัดควบคุมการไหลมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งหมด 3 จุด หมายความว่าโปรแกรมนี้มีปมเกิดขึ้นทั้งหมด 3 ปม จะพบว่าค่าความซับซ้อนจากตัววัดปมนี้จะขึ้นกับคำสั่งประเภทกระโดดข้าม เช่น GoTo Exit เท่านั้น



รูปที่ 2.8 การควบคุมการไหลจากบรรทัดการทำงานของโปรแกรม

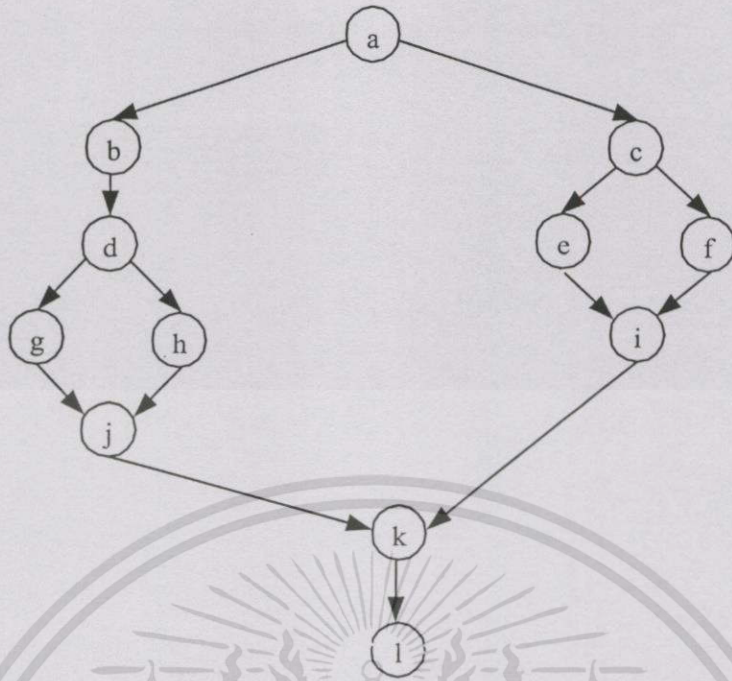
2.3.5 ตัววัดขอบเขต (Scope Metric)

ตัววัดขอบเขตนี้เป็นแนวความคิดของ Harrison และคณะ [5] คิดขึ้นเพื่อแก้ไขข้อจำกัดของตัววัดของ McCabe เนื่องจากว่าค่าไซโคลเมตริกไม่สามารถบ่งบอกอะไรเกี่ยวกับโครงสร้างการซ้อนใน โดยกำหนดให้ G เป็นกราฟย่อยของกราฟควบคุมการไหล และมีคำศัพท์ที่ควรทราบดังนี้

ขอบเขตล่าง (lower bound : lb) ของ G คือ โหนดหนึ่งใน G ที่ทุก ๆ โหนดใน G สามารถไปถึงได้

ขอบเขตล่างมากที่สุด (greatest lower bound : glb) ของ G คือ ค่าขอบเขตล่าง ที่อยู่เหนือขอบเขตล่างอื่น ๆ ของ G หรือกล่าวคือ โหนดที่อยู่บนสุดในขอบเขตล่างนั้น

ค่าความซับซ้อนจากตัววัดขอบเขต พิจารณาได้จากตัวอย่างในรูปที่ 2.9 เมื่อพิจารณาโหนด a เป็นหลัก จากโหนด a จะมีกราฟย่อย 2 กราฟ ได้แก่ กราฟด้านซ้ายเริ่มต้นที่โหนด b และกราฟด้านขวาเริ่มต้นที่โหนด c ทุก ๆ โหนดของกราฟทั้งสองนี้สามารถไปถึงโหนด k และ โหนด l ได้ ดังนั้น k และ l เป็นขอบเขตล่างของโหนด a แล้ว k เป็นขอบเขตล่างมากที่สุดของโหนด a จากนั้นนับจำนวนโหนดทั้งหมดที่อยู่ระหว่าง a กับ k จะได้ค่าความซับซ้อนของโหนด a ในกรณีนี้เท่ากับ 10 ส่วนโหนดอื่น ๆ แสดงได้ดังตารางที่ 2.3



รูปที่ 2.9 กราฟควบคุมการไหลของตัววัดขอบเขต

ตารางที่ 2.3 การหาค่าความซับซ้อนจากตัววัดขอบเขต

โหนด	ชื่อโหนดที่อยู่ระหว่างโหนด ที่พิจารณาถึง glb	ความซับซ้อน
a	b c d e f g h i j k	10
b	d g h j	4
c	e f i	3
d	g h j	3
e	i	1
f	i	1
g	j	1
h	j	1
i	k	1
j	k	1
k	l	1
l	-	0
		27

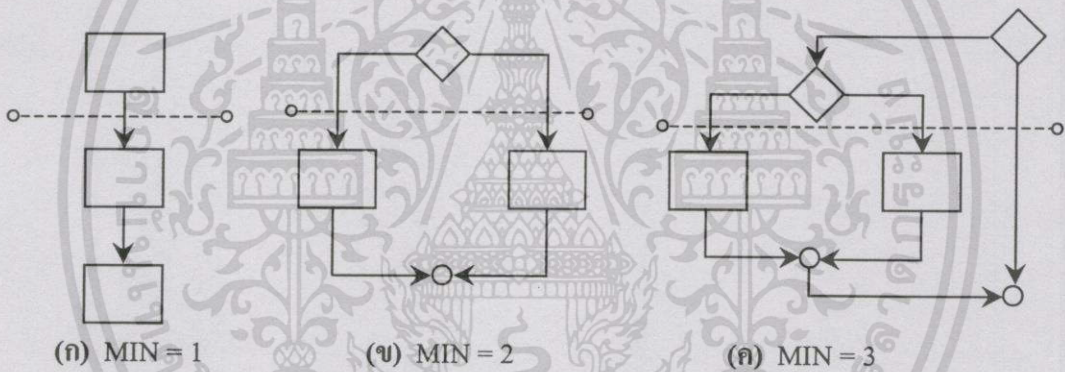
จากตารางที่ 2.3 ค่าความซับซ้อนรวมของกราฟในรูปที่ 2.9 คือ ผลรวมค่าความซับซ้อนทั้งหมดจากโหนดที่ a ถึง l ซึ่งมีค่าเท่ากับ 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพในการวัดของตัววัดนี้จะดีเฉพาะกรณีที่โครงสร้างของโปรแกรมนั้นเป็นแบบซ้อนใน และเมื่อนำโปรแกรมที่ประกอบด้วยกราฟควบคุมการไหลย่อยแบบเดียวกันแต่มีการทำงานแบบเรียงลำดับจะพบว่าค่าความซับซ้อนที่วัดได้จากตัววัดนี้ กราฟควบคุมการไหลที่มีการทำงานแบบเรียงลำดับกลับให้ค่าความซับซ้อนมากกว่าซึ่งขัดกับความเป็นจริง

2.3.6 ตัววัดของ Chen (Chen's Metric)

Edward T. Chen [11] ได้เสนอระบบที่เรียกว่า Maximal Intersect Number หรือ MIN ซึ่ง เป็นอีกวิธีที่หาค่าความซับซ้อนจากกราฟควบคุมการไหล โดยจะทำการวัดโครงสร้างการซ้อนใน (Nesting Structure) ของคำสั่งประเภทการตัดสินใจและวนซ้ำ การคำนวณตัววัดแบบนี้ โหนดเข้า (Entry Node) และโหนดออก (Exit Node) ของกราฟควบคุมการไหลต้องเชื่อมโยงกันเป็นกราฟ ต่อเนื่อง (Connected Graph) และสามารถแบ่งกราฟได้หลาย ๆ ส่วน จากนั้นลากเส้นตัดผ่านเข้าไป ยังแต่ละส่วน ค่า MIN ย่อยในแต่ละส่วนจะเท่ากับจำนวนจุดตัดที่มากที่สุด เมื่อลากเส้นตัดผ่านจาก ซ้ายไปขวาของกราฟควบคุมการไหลย่อยนั้น ดังรูปที่ 2.10



รูปที่ 2.10 การหาค่า MIN จากกราฟควบคุมการไหล

จากรูปที่ 2.10 (ก) วัดค่า MIN ได้เท่ากับ 1 รูปที่ 2.10 (ข) วัดค่า MIN ได้เท่ากับ 2 และรูปที่ 2.10 (ค) วัดค่า MIN ได้เท่ากับ 3

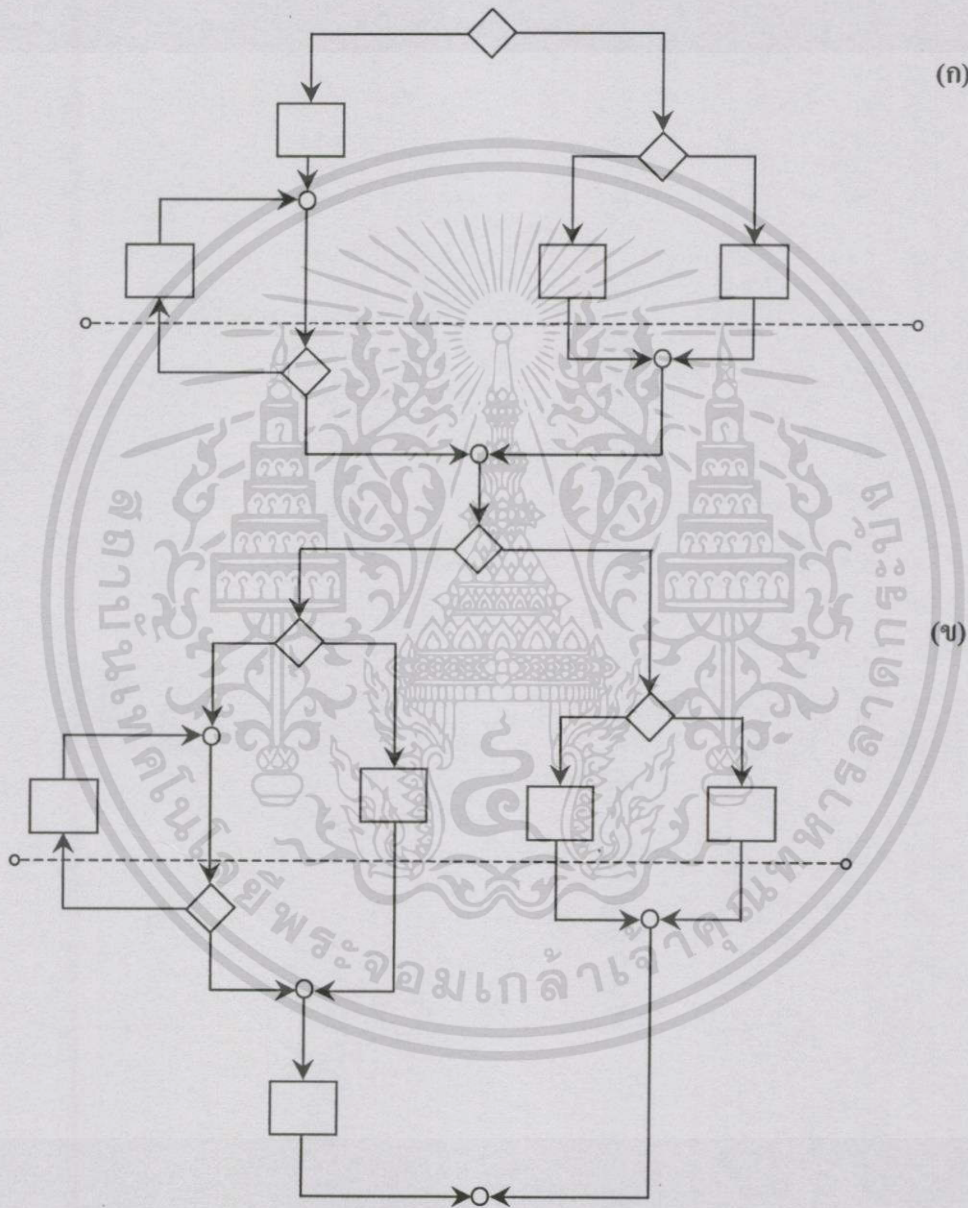
Chen ได้กำหนดสูตรการคำนวณหาค่า MIN รวมไว้ดังนี้

$$\text{MIN} = c - (2 \times s) + 2 \quad (2.6)$$

โดยที่ c คือ ผลรวมของ MIN ย่อยในแต่ละส่วน

s คือ จำนวนของส่วนย่อยที่แบ่งได้ทั้งหมดในกราฟควบคุมการไหล

การวัดความซับซ้อนจากตัววัดของ Chen แสดงได้ดังรูปที่ 2.11 ซึ่งพบจำนวนบริเวณย่อยที่แบ่งได้ทั้งหมดในกราฟควบคุมการไหล (s) รูปที่ 2.11 เท่ากับ 2 ได้แก่ บริเวณย่อย (ก) และบริเวณย่อย (ข) ส่วนผลรวมของ MIN ย่อยในแต่ละบริเวณ (c) เท่ากับ 9 คือ ค่า MIN ของบริเวณย่อย (ก) เท่ากับ 4 และค่า MIN ของบริเวณย่อย (ข) เท่ากับ 5 จากสูตรการหาค่าความซับซ้อนของ Chen $MIN = c - (2 \times s) + 2$ จะได้ MIN ของกราฟควบคุมการไหลรูปที่ 2.11 คือ $9 - (2 \times 2) + 2 = 7$



รูปที่ 2.11 กราฟควบคุมการไหลตัวอย่างการวัดของ Chen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อสังเกตบางประการเกี่ยวกับตัววัดของ Chen คือ [2]

1. โครงสร้างของโปรแกรมแบบลำดับ (Sequence) จะไม่มีผลต่อการเพิ่มขึ้นหรือลดลงของค่า MIN
2. สำหรับโปรแกรมที่ประกอบด้วยคำสั่งประเภทตัดสินใจและลำดับ ค่า MIN ย่อยของโปรแกรมจะเท่ากับผลรวมของคำสั่งประเภทตัดสินใจวทหนึ่ง ในทำนองเดียวกันกับการวัดของ McCabe

2.3.7 ตัววัดของ McClure (McClure's Metric)

ตัววัดของ McClure [12] จะพิจารณาจากจำนวนการเปรียบเทียบและจำนวนตัวแปรควบคุมที่ถูกอ้างอิงภายในโมดูล แล้วนำไปคำนวณค่าความซับซ้อนได้จากสูตร

$$C(m) = C + V \quad (2.7)$$

โดยที่ C คือ จำนวนของการเปรียบเทียบภายในโมดูล m

V คือ จำนวนของตัวแปรควบคุมที่ถูกอ้างอิงภายในโมดูล m

จากคำสั่ง Do While ((A = B) And (C = 1)) เมื่อพิจารณาจำนวนของการเปรียบเทียบและตัวแปรควบคุม จะได้จำนวนครั้งการเปรียบเทียบเท่ากับ 2 ได้แก่ (A = B) และ (C = 1) จำนวนตัวแปรควบคุมเท่ากับ 3 ได้แก่ A B และ C (ค่าคงที่ 1 ไม่นับ)

การหาค่าความซับซ้อนจากตัววัดของ McClure แสดงได้ดังตารางที่ 2.4 โดยพิจารณาจากโปรแกรมในรูปที่ 2.12 ซึ่งนับจำนวนการเปรียบเทียบของโปรแกรมรูปที่ 2.12 (ก) และ (ข) ได้เท่ากับ 3 ซึ่งคำสั่งที่มีการเปรียบเทียบของโปรแกรมรูปที่ 2.12 (ก) ได้แก่ (t = M) (t = 2) และ (t = 3) ส่วนคำสั่งที่มีการเปรียบเทียบของโปรแกรมรูปที่ 2.12 (ข) ได้แก่ (t1 = M) (t2 = 2) และ (t3 = 3) จำนวนตัวแปรควบคุมของโปรแกรมรูปที่ 2.12 (ก) เท่ากับ 2 ได้แก่ t และ M จำนวนส่วนตัวแปรควบคุมของโปรแกรมรูปที่ 2.12 (ข) เท่ากับ 4 ได้แก่ t1 t2 t3 และ M ดังนั้นผลสรุปค่าความซับซ้อนโดยตัววัดของ McClure สรุปได้ดังตารางที่ 2.4

```

If t = M Then
    x = 1
ElseIf t = 2 Then
    x = 2
ElseIf t = 3 Then
    x = 3
Else
    x = 4
End If

```

(ก)

```

If t1 = M Then
    x = 1
ElseIf t2 = 2 Then
    x = 2
ElseIf t3 = 3 Then
    x = 3
Else
    x = 4
End If

```

(ข)

รูปที่ 2.12 โปรแกรมตัวอย่างการวัดของ McClure

ตารางที่ 2.4 ผลสรุปค่าความซับซ้อนโดยตัววัดของ McClure

โปรแกรม	จำนวนการเปรียบเทียบ (C)	จำนวนตัวแปรควบคุม (V)	C(m)
(ก)	3	2	$3 + 2 = 5$
(ข)	3	4	$3 + 4 = 7$

จากตารางที่ 2.4 พบว่าโปรแกรม (ก) มีความซับซ้อนเท่ากับ 5 ส่วนโปรแกรม (ข) มีความซับซ้อนเท่ากับ 7 ดังนั้นสรุปได้ว่าโปรแกรม (ก) มีความซับซ้อนน้อยกว่าโปรแกรม (ข)

จากตัววัดนี้สังเกตได้ว่า ถ้าตัวแปรน้อยค่าความซับซ้อนจะน้อยกว่าโปรแกรมที่มีตัวแปรมากกว่า ซึ่งขัดแย้งกับหลักการเขียนโปรแกรมที่ดีควรที่จะกำหนดค่าคงที่ที่ใช้เป็นตัวแปร

2.3.8 ตัววัดของ Henry และ Kafura (Henry and Kafura's Metric)

Henry และ Kafura [10] ได้นำเสนอตัววัดความซับซ้อนโดยพิจารณาจากการไหลของข้อมูล และได้นิยามเกี่ยวกับการไหลของข้อมูลไว้ดังนี้

นิยาม 1 แฟนอิน (fan-in) ของโมดูล A หมายถึง จำนวนโครงสร้างข้อมูลที่ A รับข้อมูลเข้า

นิยาม 2 แฟนเอาท์ (fan-out) ของโมดูล A หมายถึง จำนวนของโครงสร้างข้อมูลที่มีการแก้ไขจาก A

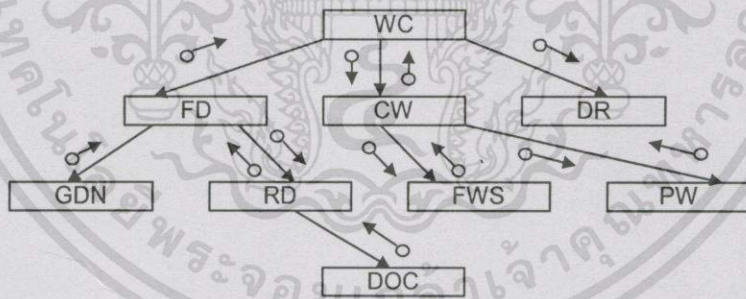
Henry และ Kafura ได้เสนอวิธีการวัดความซับซ้อนในการไหลของข้อมูลไว้ดังนี้

$$\text{Complexity}(M) = \text{length}(M) \times (\text{fan-in}(M) \times \text{fan-out}(M))^2 \quad (2.8)$$

โดยที่ length(M) แทนความยาวของโมดูล M ส่วนเทอมของ fan-in × fan-out แสดงถึงจำนวนเส้นทาง (Paths) ที่เป็นไปได้จากแหล่งข้อมูลเข้าไปยังปลายทางของข้อมูลออก การถ่วงน้ำหนักของ fan-in และ fan-out ขึ้นอยู่กับการศึกษาที่เชื่อว่าความซับซ้อนไม่เป็นเพียงแค่ความสัมพันธ์เชิงเส้นตรงในการเชื่อมต่อกับโมดูลอื่น ๆ

ตัววัดความซับซ้อนนี้จะต้องพิจารณาโครงสร้างของโมดูลทั้งหมดก่อน ซึ่งการสร้างความสัมพันธ์ระหว่างโมดูลจะยุ่งยากในทางปฏิบัติ เนื่องจากระบบอื่น ๆ จะมีขนาดที่ใหญ่มาก เมื่อกำหนดโครงสร้างความสัมพันธ์ระหว่างโมดูลเรียบร้อยแล้วจึงจะสามารถพิจารณาการไหลของข้อมูลระหว่างโมดูลได้ ซึ่งการวิเคราะห์การไหลของข้อมูลอาจทำได้ทันทีหลังจากขั้นตอนการออกแบบ เมื่อรายละเอียดภายนอกเสร็จสมบูรณ์แต่จะวิเคราะห์ก่อนการทำโปรแกรม (Implementation) ผู้วิจัยได้แสดงตัวอย่างการวัดความซับซ้อนจากตัววัดนี้ไว้ในรูปที่ 2.13

จากรูปที่ 2.13 ส่วนแรกเป็นการแสดงโครงสร้างความสัมพันธ์ระหว่างโมดูล สัญลักษณ์แสดงการไหลเข้าและออกของข้อมูล เมื่อพิจารณาโมดูล WC พบว่ามีข้อมูลที่ไหลเข้า (fan-in) 2 ตัว ซึ่งออกมาจากโมดูล FD และ โมดูล CW ส่วนข้อมูลที่ไหลออกจากโมดูล WC มี 2 ตัวเช่นกัน ซึ่งไหลออกไปยังโมดูล CW และ โมดูล DR การพิจารณาการไหลของข้อมูลโมดูลอื่น ๆ ทำได้ในทำนองเดียวกันซึ่งแสดงไว้ในตารางรูปที่ 2.13



Module	fan-in	fan-out	$[(\text{fan-in})(\text{fan-out})]^2$	length	complexity
WC	2	2	16	30	480
FD	2	2	16	11	176
CW	3	3	27	40	1080
DR	1	0	0	23	0
GDN	0	1	0	14	0
RD	2	1	4	28	112
FWS	1	1	1	46	46
PW	1	1	1	29	29

รูปที่ 2.13 การคำนวณการวัดความซับซ้อนในการไหลของข้อมูลของ Henry และ Kafura

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.9 ตัววัดของ Shepperd และ Ince (Shepperd and Ince's Metric)

Shepperd และ Ince [14] ทำการศึกษาตัววัดของ Henry และ Kafura เขาคิดว่าความยาวของโมดูลไม่น่าเป็นปัจจัยที่ส่งผลให้การไหลของข้อมูลเกิดความซับซ้อนจึงแก้ไขค่าความซับซ้อนของ Henry และ Kafura สำหรับโมดูล M ใหม่ โดยกำหนดได้ดังนี้

$$\text{complexity}(M) = (\text{fan-in}(M) \times \text{fan-out}(M))^2 \quad (2.9)$$

Shepperd และ Ince กล่าวว่า การวัดของเขามีประสิทธิภาพกว่าวิธีเดิม วิธีการวัดใหม่นี้จะกำจัดข้อมูลที่ไม่ชัดเจนและพิจารณาเฉพาะการไหลของข้อมูลอย่างเดียวนั้น วิธีการวัดของ Henry และ Kafura ขัดแย้งกับทฤษฎีการวัดซึ่งต้องการลักษณะของคุณสมบัติเฉพาะ Shepperd และ Ince พยายามดึงเฉพาะโครงสร้างการไหลของข้อมูลให้ตรงกับทฤษฎีการวัด จากการศึกษาเหตุผลจากหลักเกณฑ์ที่ Shepperd และ Ince ตั้งขึ้น ทดสอบได้ว่ามีขบวนการวัดสั้นขึ้นถือเป็นการพัฒนาเวลาการคำนวณได้เร็วขึ้นและเป็นการวัดเฉพาะการไหลของข้อมูลจริง ๆ

2.3.10 ตัววัดของ Oviedo (Oviedo's Metric)

ตัววัดของ Oviedo [15] พิจารณาจากกราฟควบคุมการไหลและวิธีการใช้ข้อมูล ซึ่งเป็นวิธีที่วัดการไหลของข้อมูล (Data Flow) ด้วย Oviedo นิยามการวัดความซับซ้อนไว้ดังนี้

$$C = cf + df \quad (2.10)$$

โดยที่ cf คือ จำนวนเส้นเชื่อมของกราฟควบคุมการไหล ซึ่งเป็นค่าการควบคุมการไหล df คือ ค่าการไหลของข้อมูล (Data Flow)

โดยนิยามค่า df เท่ากับผลบวกของค่า df พิจารณาตัวแปรในทุก ๆ โหนดนั้นคือ

$$df = \sum_{i=1}^{|V|} df_i \quad (2.11)$$

โดยที่ $|V|$ คือ จำนวนโหนดในกราฟควบคุมการไหล

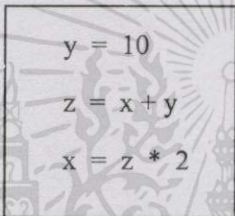
df_i คือ จำนวนตัวแปรที่เฉพาะถิ่นที่ได้รับการกำหนดค่าก่อน โหนดที่ i และมีเส้นเชื่อมโยงถึงโหนดที่ i

คำศัพท์ที่ควรทราบในการวัดค่าความซับซ้อนของ Oviedo

การกำหนดค่าตัวแปร (Variable Definition) หมายถึง ตัวแปรที่รับค่าจากคำสั่งรับค่าหรือมีการรับค่าจากคำสั่งกำหนดค่า (Assignment) หรือจากโปรแกรมย่อย เช่น $x = 10$ `interchange(x, y)` เรียกว่ามีการกำหนดค่าให้กับตัวแปร x

ตัวแปรได้รับการใช้หรืออ้างอิง หมายถึง ตัวแปรที่มีการใช้ค่าตัวแปรนั้นในคำสั่งประเภทกำหนดค่าหรือคำสั่งแสดงผลลัพธ์ เช่น `Print x` $y = x + 10$ `If x = 10 Then` จะเรียกว่าตัวแปร x ได้รับการใช้หรืออ้างอิง

ตัวแปรตีแผ่เฉพาะถิ่น (Locally Exposed Variable) หมายถึง ตัวแปรที่ได้รับการใช้หรืออ้างอิง โดยที่ไม่มีการกำหนดค่าตัวแปรไว้ก่อนในโหนดที่พิจารณา เช่น



```

y = 10
z = x + y
x = z * 2

```

เมื่อพิจารณาตัวแปรในโหนดนี้พบว่าตัวแปร y และ z เป็นตัวแปรที่ได้รับการกำหนดค่าในโหนดนี้ ส่วนตัวแปร x เป็นตัวแปรตีแผ่เฉพาะถิ่น

ตัวอย่างการหาค่าความซับซ้อนด้วยวิธีของ Oviedo จากโปรแกรมในรูปที่ 2.14 (ก) แสดงในรูปของกราฟควบคุมการไหลได้ดังรูปที่ 2.14 (ข)

จากรูปที่ 2.14 (ก) ค่า cf หรือจำนวนเส้นเชื่อมทั้งหมดเท่ากับ 5 ส่วนค่า df ย่อยในแต่ละโหนดแสดงไว้ในตารางที่ 2.5 พร้อมทั้งแสดงค่าตัวแปรตีแผ่เฉพาะถิ่นของแต่ละโหนดที่ได้รับการกำหนดค่ามาก่อน โดยที่ $x(i)$ หมายถึง การกำหนดค่าของตัวแปร x ในโหนดที่ i ซึ่งเป็นโหนดที่มาก่อนและมีเส้นเชื่อมโยงถึงโหนดที่พิจารณา

```

n = Val(Form1.Text(0).text)
x = Val(Form1.Text(1).text)
k = Val(Form1.Text(2).text)

```

```

If n = 1 Then

```

```

    x = 1

```

```

    j = 2

```

```

    m = 3

```

```

Else

```

```

    k = 1

```

```

    j = 2

```

```

End If

```

```

a = x + j - k

```

```

x = n + 5

```

(ก)

```

n = Val(Form1.Text(0).text)
x = Val(Form1.Text(1).text)
k = Val(Form1.Text(2).text)

```

โหนด 1

```

If n = 1 Then

```

โหนด 2

```

x = 1

```

```

j = 2

```

```

m = 3

```

โหนด 3

โหนด 4

```

k = 1

```

```

j = 2

```

```

a = x + j - k

```

```

x = n + 5

```

โหนด 5

(ข)

รูปที่ 2.14 ตัวอย่างโปรแกรมและกราฟควบคุมการไหลของโปรแกรมของตัววัด Oviedo

ตารางที่ 2.5 ค่าความซับซ้อนการไหลของข้อมูล

โหนดที่	ตัวแปรที่แผ่เฉพาะอื่น	การกำหนดค่าครั้งก่อน	ความซับซ้อนการไหลของข้อมูล
1	-	-	-
2	n	n(1)	1
3	-	-	-
4	-	-	-
5	x	x(1), x(3)	2
	j	j(3), j(4)	2
	k	k(1), k(4)	2
	n	n(1)	1
$\sum df_i$			8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 2.5 ผลรวมของ df ย่อย หรือ $\sum_{i=1}^k df_i$ เท่ากับ 8 ดังนั้นค่าความซับซ้อนของ

Oviedo ของโปรแกรมมีค่าเท่ากับ $5 + 8 = 13$

จากวิธีการของ Oviedo พบว่า ถ้าตัวแปรที่ใช้เป็นตัวแปรโกลบอล และมีการใช้ตัวแปรนั้น โดยไม่มีการกำหนดค่าตัวแปรนั้นไว้ในโปรแกรมที่พิจารณา ประสิทธิภาพการวัดของตัววัดนี้จะลดลง

จากการศึกษาตัววัดทั้ง 10 แบบจะพบว่านักวิจัยส่วนมากไม่สนับสนุนการวัดความซับซ้อนจากตัววัดจำนวนบรรทัด (Line of code) เนื่องจากนักวิจัยคิดว่าความซับซ้อนของโปรแกรมไม่ได้ขึ้นอยู่กับจำนวนบรรทัดของโปรแกรมแต่น่าจะขึ้นอยู่กับโครงสร้างในการเขียนโปรแกรม และจากตัววัดของ Halstead พิจารณาความซับซ้อนจากโอเปอเรเตอร์และโอเปอเรนด์เท่านั้น ซึ่งเขาเป็นผู้ให้กำหนดสาขาย่อยของวิทยาการคอมพิวเตอร์ที่ชื่อว่าตัววัดความซับซ้อนของซอฟต์แวร์

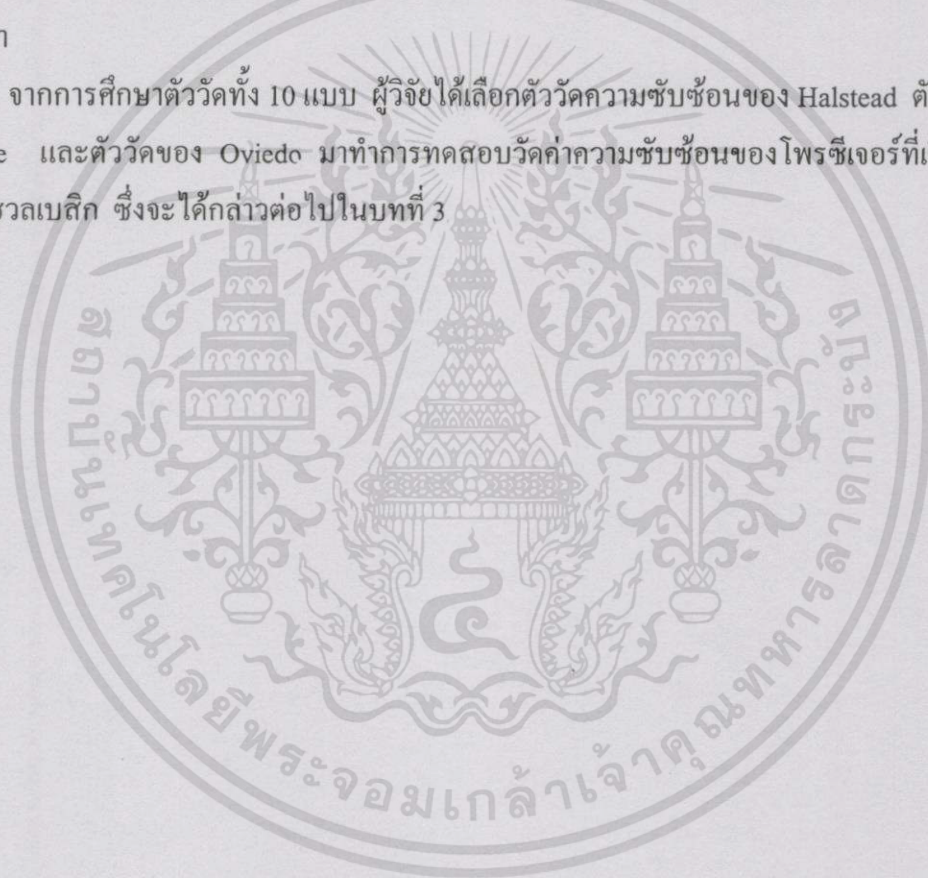
ในการวัดความซับซ้อนด้านโครงสร้าง McCabe เป็นผู้เริ่มนำเสนอแนวทางในการวัดโปรแกรมแบบโครงสร้างโดยจะพิจารณาจากกราฟควบคุมการไหล และ McCabe ยังได้นำเสนอเกณฑ์การวัดความซับซ้อนที่เหมาะสมไว้ด้วย ต่อมามีนักวิจัยหลายท่านนำหลักการของ McCabe ไปประยุกต์ใช้และกำหนดตัววัดใหม่ เช่น ตัววัด bpm ซึ่งค่าความซับซ้อนจากตัววัด bpm นี้จะขึ้นกับคำสั่งประเภทกระโดดข้าม เช่น GoTo Exit เท่านั้น ผู้วิจัยคิดว่าค่าความซับซ้อนไม่น่าจะขึ้นอยู่กับคำสั่งประเภทกระโดดข้ามเท่านั้น Harrison และคณะได้นำเสนอตัววัดขอบเขตโดยพยายามหาข้อจำกัดจากตัววัดของ McCabe ซึ่งไม่แสดงถึงการซ้อนใน แต่ประสิทธิภาพในการวัดของตัววัดนี้ยังไม่ดีนักในกรณีที่โปรแกรมนั้นมีโครงสร้างการทำงานแบบเรียงลำดับ เมื่อพิจารณาโปรแกรมที่ประกอบด้วยกราฟควบคุมการไหลย่อยแบบเดียวกันแต่มีการทำงานแบบเรียงลำดับกับแบบซ้อนใน จะพบว่าค่าความซับซ้อนที่วัดได้จากตัววัดนี้ กราฟควบคุมการไหลที่มีการทำงานแบบเรียงลำดับกลับให้ค่าความซับซ้อนมากกว่าซึ่งตามหลักการเขียนโปรแกรมแบบโครงสร้างโปรแกรมที่มีการทำงานแบบเรียงลำดับน่าที่จะให้ค่าความซับซ้อนน้อยกว่าโปรแกรมที่มีการทำงานแบบซ้อนใน ส่วนตัววัดของ Chen ได้นำเสนอระบบที่เรียกว่า Maximal Intersect Number หรือ MIN ซึ่งเป็นอีกวิธีที่หาค่าความซับซ้อนจากกราฟควบคุมการไหล ตัววัดนี้ยังคงให้ค่าความซับซ้อนเป็นไปไม่แน่นอนเดียวกันกับตัววัดของ McCabe สำหรับโปรแกรมที่ประกอบด้วยคำสั่งประเภทตัดสินใจและลำดับ ค่า MIN ย่อยของโปรแกรมจะเท่ากับผลรวมของคำสั่งประเภทตัดสินใจบวกหนึ่ง ตัววัดของ McClure จะพิจารณาจากจำนวนการเปรียบเทียบและจำนวนตัวแปรควบคุมที่ถูกอ้างอิงภายในโมดูล จากตัววัดนี้สังเกตได้ว่าถ้าตัวแปรน้อยค่าความซับซ้อนจะน้อยกว่าโปรแกรมที่มีตัวแปรมากกว่า ซึ่งขัดแย้งกับหลักการเขียนโปรแกรมที่ดีควรที่จะกำหนดค่าคงที่ที่ใช้เป็นตัวแปร

ตัววัดของ Henry และ Kafura จัดอยู่ในลักษณะของตัววัดข้อมูล ซึ่งพิจารณาความซับซ้อนจากการไหลของข้อมูลและความยาวของโปรแกรม ตัววัดความซับซ้อนนี้จะต้องพิจารณาโครง

สร้างของโมดูลทั้งหมดก่อน ซึ่งการสร้างความสัมพันธ์ระหว่างโมดูลจะยุ่งยากในทางปฏิบัติ เนื่องจากจากระบบอื่น ๆ จะมีขนาดใหญ่มาก เมื่อกำหนดโครงสร้างความสัมพันธ์ระหว่างโมดูลเรียบร้อยแล้วจึงจะสามารถพิจารณาการไหลของข้อมูลระหว่างโมดูลได้ ต่อมา Shepperd และ Ince ได้นำเสนอตัววัดเดิมของ Henry และ Kafura แต่ไม่พิจารณาความยาวของโปรแกรม เนื่องจากเขาคิดว่าความยาวของโปรแกรมไม่ส่งผลให้การไหลของข้อมูลซับซ้อนเพิ่มขึ้น

ตัววัดสุดท้ายเป็นตัววัดแบบผสมของ Oviedo ซึ่ง Oviedo พิจารณาความซับซ้อนจากโครงสร้างควบคุมการไหล และโครงสร้างการไหลของข้อมูล การวัดความซับซ้อนจากโครงสร้างยังอ้างอิงจากตัววัดของ McCabe และประสิทธิภาพในการวัดของตัววัดนี้จะลดลง ถ้าตัวแปรที่ใช้เป็นตัวแปรโกลบอล และมีการใช้ตัวแปรนั้นโดยไม่มีกำหนดค่าตัวแปรนั้นไว้ในโปรแกรมที่พิจารณา

จากการศึกษาตัววัดทั้ง 10 แบบ ผู้วิจัยได้เลือกตัววัดความซับซ้อนของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo มาทำการทดสอบวัดค่าความซับซ้อนของโปรแกรมที่เขียนจากภาษาวิซวลเบสิก ซึ่งจะได้อีกต่อไปในบทที่ 3



บทที่ 3

ปัจจัยและการทดสอบวัดความซับซ้อนจาก ตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo

ในบทนี้ นำเอาตัววัดที่ได้จากการศึกษาในบทที่ 2 มาพิจารณาปัจจัยตามโครงสร้างภาษา
วิชาวเบสิก เพื่อทำการทดสอบวัดค่าความซับซ้อนของโปรแกรมระบบงานที่พัฒนาขึ้นด้วยภาษา
วิชาวเบสิกด้วยตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo ตัววัดของ
Halstead ในงานวิจัยนี้เราจะพิจารณาความซับซ้อนจากค่าความพยายามของโปรแกรมเท่านั้น

3.1 ปัจจัยตัววัดของ Halstead

ปัจจัยตัววัดของ Halstead มีดังนี้

1. โอเปอเรเตอร์ แบ่งเป็น 3 กลุ่มดังนี้

1. โอเปอเรเตอร์พื้นฐาน แสดงในตารางที่ 2.1
2. โอเปอเรเตอร์ที่เป็นคำสงวน แสดงในภาคผนวก ก. และในภาษาวิชาวเบสิกมี
การใช้สัญลักษณ์แทนประเภทของตัวแปรดังแสดงในตารางที่ 3.1 ซึ่งจะนับเป็นโอเปอเรเตอร์ด้วย
3. โอเปอเรเตอร์พิเศษ ได้แก่ ชื่อซบรูทินซึ่งจะปรากฏหลังคำสั่ง Call และชื่อ
ฟังก์ชันแสดงในภาคผนวก ข.

ตัวแปรที่เกี่ยวกับโอเปอเรเตอร์ ได้แก่ n_1 แทนจำนวนโอเปอเรเตอร์ทั้งหมดที่ไม่ซ้ำกัน
ในโปรแกรม (Unique Operators) และ N_1 แทนจำนวนโอเปอเรเตอร์ทั้งหมดในโปรแกรม
(Total Operators)

ตารางที่ 3.1 ประเภทและสัญลักษณ์ที่ใช้แทนประเภทของตัวแปร

ประเภท	สัญลักษณ์
Integer	%
Long	&
Single	!
Double	#
Currency	@
String	\$

2. โอเปอเรนด์ ได้แก่ ตัวแปรและค่าคงที่ที่ใช้ในโปรแกรม ไม่พิจารณาส่วนที่เป็นการประกาศค่าตัวแปรภายใต้คำสั่ง Dim และ Const ค่าคงที่ที่เป็นค่าสวจน ได้แก่ True, False, Me, Null นับเป็นโอเปอเรนด์ด้วย

ตัวแปรที่เกี่ยวกับโอเปอเรนด์ ได้แก่ n_2 แทนจำนวนโอเปอเรนด์ทั้งหมดที่ไม่ซ้ำกันในโปรแกรม (Unique Operands) และ N_2 แทนจำนวนโอเปอเรนด์ทั้งหมดในโปรแกรม (Total Operands)

3.2 ปัจจัยตัววัดของ McCabe

ปัจจัยตัววัดของ McCabe มีดังนี้

1. โหนด

M McCabe กำหนดตัวแปร n แทนจำนวนโหนดของกราฟควบคุมการไหล ดังนั้นเราจึงจำเป็นต้องพิจารณาว่าแต่ละ โหนดจะประกอบด้วยคำสั่งใดบ้าง ซึ่งผู้วิจัยจะกำหนดขอบเขตของโหนดในการวิจัยนี้เป็นการกำหนดตามข้อกำหนดของ Oviedo เพื่อนำไปใช้ในการพิจารณาค่า cf กับตัววัดของ Oviedo ได้ด้วย ซึ่ง Oviedo กำหนดสมาชิกของแต่ละโหนดจะประกอบด้วย [12]

1. ประโยคคำสั่ง Sub
2. ประโยคคำสั่ง End Sub
3. ประโยคคำสั่งประเภทตัดสินใจพร้อมด้วยคำหลักที่เกี่ยวข้อง ได้แก่ If , ElseIf , While , Until , For , Select Case

4. กลุ่มประโยคของคำสั่งที่มีการทำงานแบบเรียงลำดับ ซึ่งกำหนดดังนี้

4.1 ประโยคคำสั่งแรกที่เป็นสมาชิกในกลุ่ม คือ

4.1.1 ประโยคคำสั่งแรกหลังจากคำสั่ง Sub

4.1.2 ป้ายชื่อ (Label) ซึ่งทำงานตามคำสั่ง GoTo

4.1.3 ประโยคคำสั่ง Then

4.1.4 ประโยคคำสั่ง Else

4.1.5 ประโยคคำสั่งแรกหลังจากเงื่อนไข GoTo เช่น ประโยคคำสั่ง S2 ใน

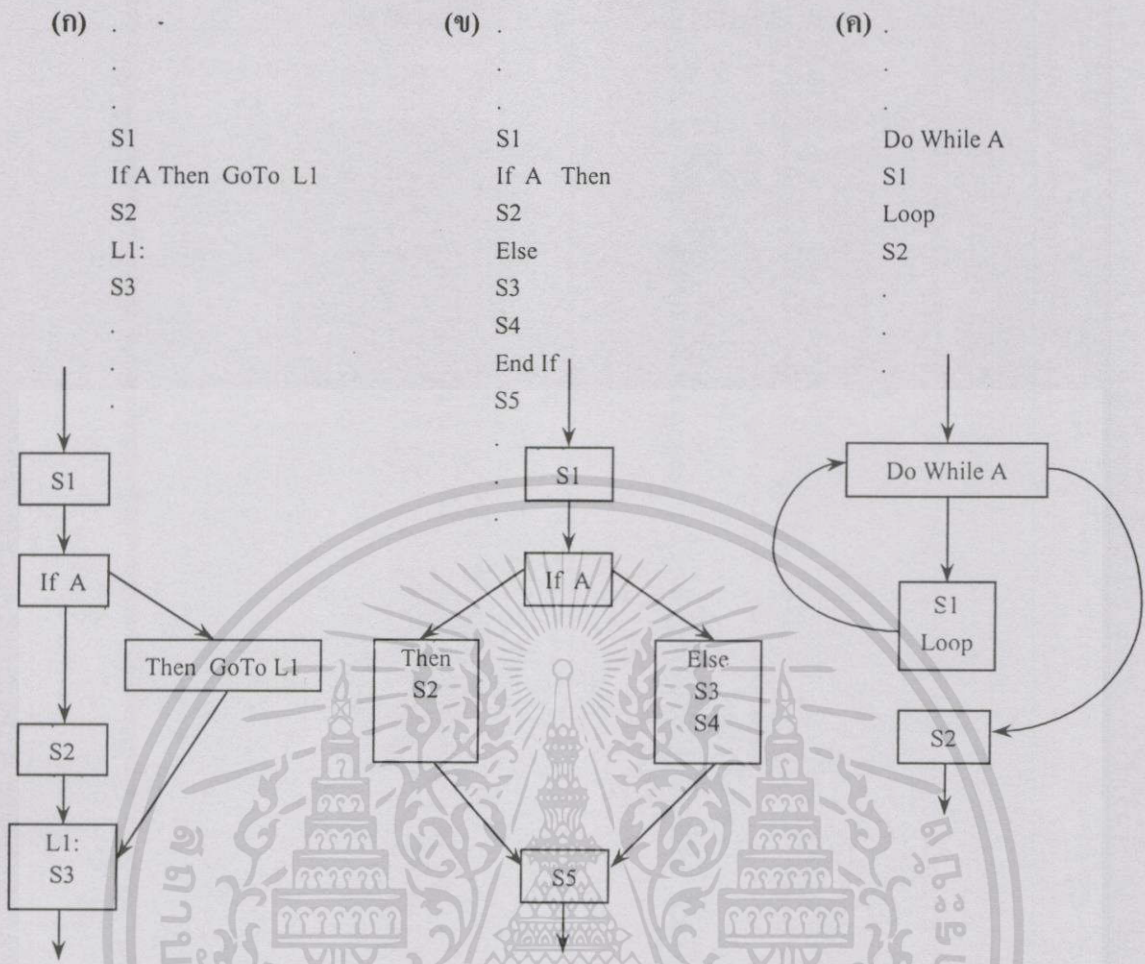
รูปที่ 3.1 (ก)

4.1.6 ประโยคคำสั่งที่มีโหนดตั้งแต่ 2 โหนดไปถึงได้ เช่น ประโยคคำสั่ง S5 ในรูปที่ 3.1 (ข)

4.1.7 ประโยคคำสั่งที่อยู่ถัดมาจากประโยคคำสั่ง Do {While/Until} เช่น ประโยคคำสั่ง S1 และ S2 ในรูปที่ 3.1 (ค)

4.1.8 บรรทัดแรกหลังจากประโยคคำสั่ง Loop {While/Until} หรือ Next หรือ Wend

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1

- (ก) ขอบเขตของโหนดและกราฟควบคุมการไหลของโครงสร้างโปรแกรม If...Then
- (ข) ขอบเขตของโหนดและกราฟควบคุมการไหลของโครงสร้างโปรแกรม If...Else
- (ค) ขอบเขตของโหนดและกราฟควบคุมการไหลของโครงสร้างโปรแกรม Do While

- 4.2 ประโยคคำสั่งสุดท้ายที่เป็นสมาชิกในกลุ่ม คือ
- 4.2.1 ประโยคคำสั่ง Loop หรือ Wend หรือ Next
 - 4.2.2 ประโยคคำสั่งที่ประมวลผลแบบไม่มีเงื่อนไข หรือ คำสั่ง GoTo
 - 4.2.3 ประโยคคำสั่งที่อยู่ถัดจากประโยคคำสั่งในข้อ 4.1 และ ข้อ 1-4 ข้างต้น

2. เส้นเชื่อมระหว่างโหนด

McCabe กำหนดตัวแปร e แทนเส้นเชื่อมระหว่างโหนด ผู้วิจัยได้กำหนดลักษณะเส้นเชื่อมของแต่ละโหนดไว้ 4 แบบ คือ

1. โหนดที่มีเส้นเชื่อมออกจากตัวเองหนึ่งเส้น คำสั่งสุดท้ายในโหนดนั้นที่ไม่ใช่คำสั่งคำสั่งประเภทการตัดสินใจ หรือเป็นคำสั่ง GoTo ที่ไม่มีการตัดสินใจ หรือคำสั่งหยุดการทำงานจากการวนรอบ เช่น Exit For , Exit Do , Exit Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โหนดที่มีเส้นเชื่อมออกจากตัวเอง 2 เส้น ได้แก่ โหนดที่เป็นประโยคคำสั่งประเภท การตัดสินใจ ได้แก่ If , ElseIf , Do {While/Until} , Loop {While/Until} , For และ On Error GoTo

3. โหนดที่มีเส้นเชื่อมออกจากตัวเองอย่างน้อย 2 เส้น ได้แก่ โหนดที่เป็นประโยคคำสั่ง Select Case จำนวนเส้นเชื่อมที่ออกจาก โหนด Select Case จะเท่ากับจำนวน Case ที่ปรากฏ

4. โหนดที่ไม่มีเส้นเชื่อมออกจากตัวเอง ได้แก่ โหนดที่มีประโยคคำสั่ง End Sub หรือ เรียกว่าโหนดสิ้นสุด

3.3 ปัจจัยตัววัดของ Oviedo

ปัจจัยตัววัดของ Oviedo มีดังนี้

1. ตัวแปรพิเศษเฉพาะถิ่น

ตัวแปรพิเศษเฉพาะถิ่น หมายถึง ตัวแปรที่ได้รับการใช้หรืออ้างอิง โดยที่ตัวแปรนั้นไม่ได้มีการกำหนดค่าไว้ก่อนในโหนดที่พิจารณา

การกำหนดค่าตัวแปร หมายถึง มีตัวแปรได้รับค่าจากคำสั่งรับค่าหรือมีการรับค่าจากคำสั่ง กำหนดค่า (Assignment) หรือจากโปรแกรมย่อย เช่น $x = 10$ interchange(x ,y) เรียกว่ามีการ กำหนดค่าให้กับตัวแปร x

Oviedo ได้กำหนดตัวแปร df แทนจำนวนตัวแปรพิเศษเฉพาะถิ่นที่ได้รับการกำหนดค่าไว้ ก่อนหน้าโหนดที่พิจารณา และมีเส้นเชื่อมโยงถึงโหนดที่พิจารณา

2. เส้นเชื่อมระหว่างโหนด

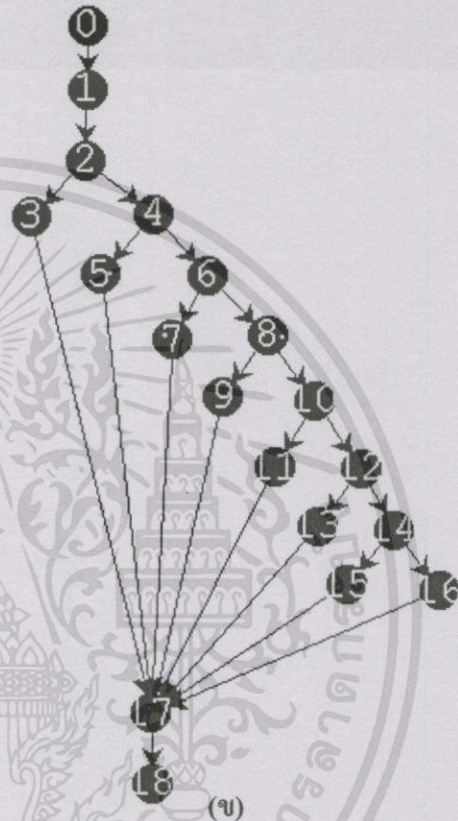
Oviedo กำหนดตัวแปร cf แทนจำนวนเส้นเชื่อมของกราฟควบคุมการไหล ซึ่งจะมีค่า เท่ากับค่า e จากตัววัดของ McCabe ดังนั้นจึงกำหนดได้ว่า $cf = e$

3.4 การทดสอบวัดความซับซ้อนของโปรแกรม

จากการพิจารณาปัจจัยตามโครงสร้างภาษาวิหวลเบสิกของตัววัดทั้ง 3 แบบในข้อที่ 3.1 ถึง 3.4 ผู้วิจัยจึงได้ทำการทดสอบโปรแกรมเพื่อหาค่าของปัจจัยที่เกี่ยวกับตัววัดทั้ง 3 แบบ โดยทำการทดสอบโปรแกรมทั้งหมด 5 โปรแกรมดังต่อไปนี้

โปรแกรมทดสอบที่ 1

Line	Node	
1	0	Sub Text1_Change()
2	1	I = Val(Text1)
3	2	If I = 1 Then
4	3	Print "วันอาทิตย์"
5	4	Elseif I = 2 Then
6	5	Print "วันจันทร์"
7	6	Elseif I = 3 Then
8	7	Print "วันอังคาร"
9	8	Elseif I = 4 Then
10	9	Print "วันพุธ"
11	10	Elseif I = 5 Then
12	11	Print "วันพฤหัสบดี"
13	12	Elseif I = 6 Then
14	13	Print "วันศุกร์"
15	14	Elseif I = 7 Then
16	15	Print "วันเสาร์"
17	16	Else
18	16	Print "เลขไม่ถูกต้อง"
19	17	End If
20	18	End Sub



รูปที่ 3.2 (ก) โปรแกรมทดสอบที่ 1

(ข) กราฟควบคุมการไหลของโปรแกรมทดสอบที่ 1

จากรูปที่ 3.2 (ก) แสดงโปรแกรมทดสอบที่ 1 สามารถกำหนดขอบเขตของโหนดได้ดังนี้ โปรแกรมบรรทัดที่ 1 เป็นสมาชิกในโหนดที่ 0 โปรแกรมบรรทัดที่ 2 เป็นสมาชิกในโหนดที่ 1 โปรแกรมบรรทัดที่ 3 เป็นสมาชิกในโหนดที่ 2 โปรแกรมบรรทัดที่ 4 เป็นสมาชิกในโหนดที่ 3 โปรแกรมบรรทัดที่ 5 เป็นสมาชิกในโหนดที่ 4 โปรแกรมบรรทัดที่ 6 เป็นสมาชิกในโหนดที่ 5 โปรแกรมบรรทัดที่ 7 เป็นสมาชิกในโหนดที่ 6 โปรแกรมบรรทัดที่ 8 เป็นสมาชิกในโหนดที่ 7 โปรแกรมบรรทัดที่ 9 เป็นสมาชิกในโหนดที่ 8 โปรแกรมบรรทัดที่ 10 เป็นสมาชิกในโหนดที่ 9 โปรแกรมบรรทัดที่ 11 เป็นสมาชิกในโหนดที่ 10 โปรแกรมบรรทัดที่ 12 เป็นสมาชิกในโหนดที่ 11 โปรแกรมบรรทัดที่ 13 เป็นสมาชิกในโหนดที่ 12 โปรแกรมบรรทัดที่ 14 เป็นสมาชิกในโหนดที่ 13

โปรแกรมบรรทัดที่ 15 เป็นสมาชิกในโหนดที่ 14 โปรแกรมบรรทัดที่ 16 เป็นสมาชิกในโหนดที่ 15 โปรแกรมบรรทัดที่ 17 และ 18 เป็นสมาชิกในโหนดที่ 16 โปรแกรมบรรทัดที่ 19 เป็นสมาชิกในโหนดที่ 17 และโปรแกรมบรรทัดที่ 20 เป็นสมาชิกในโหนดที่ 18 ดังนั้นจึงสามารถสร้างกราฟควบคุมการไหลได้ดังรูปที่ 3.2 (ข) จำนวนโหนดที่ได้เท่ากับ 19 จำนวนเส้นเชื่อมที่ได้เท่ากับ 25 และสามารถแสดงจำนวนโอเปอเรเตอร์และโอเปอเรนด์ได้ดังตารางที่ 3.2 ส่วนตัวแปรที่ได้รับการกำหนดค่าและตัวแปรตีแผ่เฉพาะถิ่นแสดงได้ดังตารางที่ 3.3

ตารางที่ 3.2 จำนวนโอเปอเรเตอร์และโอเปอเรนด์ของโปรแกรมทดสอบที่ 1

โอเปอเรเตอร์	จำนวน	โอเปอเรนด์	จำนวน
Sub	1	I	8
()	2	Text1	1
=	8	1	1
Val	1	"วันอาทิตย์"	1
If	1	2	1
Then	7	"วันจันทร์"	1
Print	8	3	1
""	8	"วันอังคาร"	1
Elseif	6	4	1
Else	1	"วันพุธ"	1
End If	1	5	1
End Sub	1	"วันพฤหัสบดี"	1
		6	1
		"วันศุกร์"	1
		7	1
		"วันเสาร์"	1
		"เลขไม่ถูกต้อง"	1
$n_1 = 12$	$N_1 = 45$	$n_2 = 17$	$N_2 = 24$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 ตัวแปรตีแผ่เฉพาะถิ่น การกำหนดค่าตัวแปรครั้งก่อน และความซับซ้อนจากการไหลของข้อมูล ของโปรแกรมทดสอบที่ 1

โหนดที่	ตัวแปรตีแผ่เฉพาะถิ่น	การกำหนดค่าตัวแปรครั้งก่อน	df _i
2	I	I(1)	1
4	I	I(1)	1
6	I	I(1)	1
8	I	I(1)	1
10	I	I(1)	1
12	I	I(1)	1
14	I	I(1)	1
df			7

ปัจจัยทั้งหมดที่ได้ของโปรแกรมทดสอบที่ 1 สรุปได้ดังนี้

$$\begin{aligned}
 n &= 19, & e &= cf = 25 \\
 n_1 &= 12, & n_2 &= 17 \\
 N_1 &= 45, & N_2 &= 24 \\
 df &= 7
 \end{aligned}$$

เมื่อนำค่าที่ได้แทนในสูตรตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo จะได้ค่าความซับซ้อนดังแสดงในตารางที่ 3.4

ตารางที่ 3.4 ค่าความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo ของโปรแกรมทดสอบที่ 1

ตัววัด	ความซับซ้อน
Halstead	$12 \times 24 \times (45 + 24) \log_2(12 + 17) / 2 \times 17 = 2,839.347$
McCabe	$25 - 19 + 2 = 8$
Oviedo	$25 + 7 = 32$

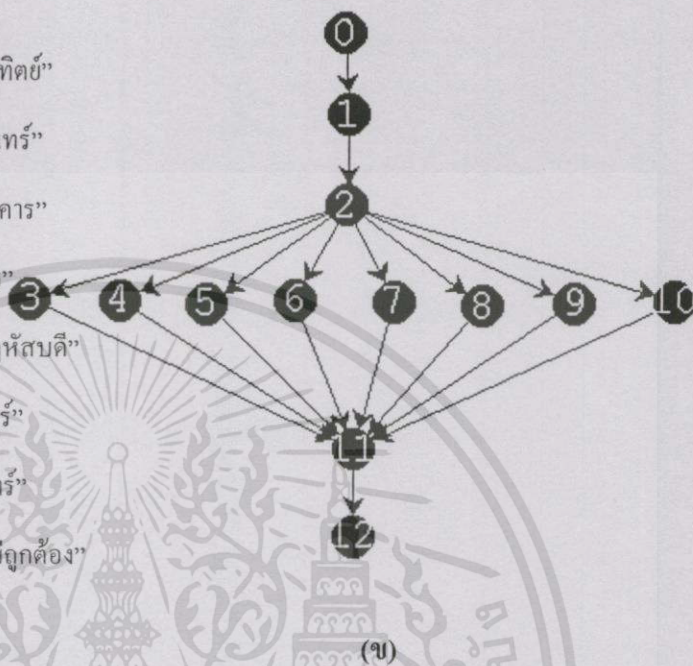
โปรแกรมทดสอบที่ 1 แสดงการเขียนโปรแกรมโครงสร้าง nested-If ผลการทดสอบวัดความซับซ้อนของโปรแกรมทดสอบที่ 1 จากตารางที่ 3.4 ค่าความซับซ้อนจากตัววัดของ Halstead มีค่าเท่ากับ 2,839.347 ส่วนค่าความซับซ้อนจากตัววัดของ McCabe มีค่าเท่ากับ 8 และค่าความซับซ้อนจากตัววัดของ Oviedo เท่ากับ 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมทดสอบที่ 2

Line Node

1	0	Sub Text1_Change()
2	1	I = Val(Text1)
3	2	Select Case I
4	3	Case 1
5	3	Print "วันอาทิตย์"
6	4	Case 2
7	4	Print "วันจันทร์"
8	5	Case 3
9	5	Print "วันอังคาร"
10	6	Case 4
11	6	Print "วันพุธ"
12	7	Case 5
13	7	Print "วันพฤหัสบดี"
14	8	Case 6
15	8	Print "วันศุกร์"
16	9	Case 7
17	9	Print "วันเสาร์"
18	10	Case Else
19	10	Print "เลขไม่ถูกต้อง"
20	11	End Select
21	12	End Sub



รูปที่ 3.3

(ก) โปรแกรมทดสอบที่ 2

(ข) กราฟควบคุมการไหลของโปรแกรมทดสอบที่ 2

จากรูปที่ 3.3 (ก) แสดงโปรแกรมทดสอบที่ 2 สามารถกำหนดขอบเขตของโหนดได้ดังนี้ โปรแกรมบรรทัดที่ 1 เป็นสมาชิกในโหนดที่ 0 โปรแกรมบรรทัดที่ 2 เป็นสมาชิกในโหนดที่ 1 โปรแกรมบรรทัดที่ 3 เป็นสมาชิกในโหนดที่ 2 โปรแกรมบรรทัดที่ 4 และ 5 เป็นสมาชิกในโหนดที่ 3 โปรแกรมบรรทัดที่ 6 และ 7 เป็นสมาชิกในโหนดที่ 4 โปรแกรมบรรทัดที่ 8 และ 9 เป็นสมาชิกในโหนดที่ 5 โปรแกรมบรรทัดที่ 10 และ 11 เป็นสมาชิกในโหนดที่ 6 โปรแกรมบรรทัดที่ 12 และ 13 เป็นสมาชิกในโหนดที่ 7 โปรแกรมบรรทัดที่ 14 และ 15 เป็นสมาชิกในโหนดที่ 8 โปรแกรมบรรทัดที่ 16 และ 17 เป็นสมาชิกในโหนดที่ 9 โปรแกรมบรรทัดที่ 18 และ 19 เป็นสมาชิกในโหนดที่ 10 โปรแกรมบรรทัดที่ 20 เป็นสมาชิกในโหนดที่ 11 และ โปรแกรมบรรทัดที่ 21 เป็นสมาชิกในโหนดที่ 12 ดังนั้นจึงสามารถสร้างกราฟควบคุมการไหลได้ดังรูปที่ 3.3 (ข) จำนวนโหนดที่ได้เท่ากับ 13 จำนวนเส้นเชื่อมที่ได้เท่ากับ 19 และสามารถแสดงจำนวนโอเปอเรเตอร์และโอเปอเรนด์ได้ดังตารางที่ 3.5 ส่วนตัวแปรที่ได้รับการกำหนดค่าและตัวแปรที่เฉพาะถิ่นแสดงได้ดังตารางที่ 3.6

ตารางที่ 3.5 จำนวนโอเปอเรเตอร์และโอเปอเรนด์ของโปรแกรมทดสอบที่ 2

โอเปอเรเตอร์	จำนวน	โอเปอเรนด์	จำนวน
Sub	1	I	2
()	2	Text1	1
=	1	1	1
Val	1	"วันอาทิตย์"	1
Select Case	1	2	1
Case	8	"วันจันทร์"	1
Print	8	3	1
""	8	"วันอังคาร"	1
Else	1	4	1
End Select	1	"วันพุธ"	1
End Sub	1	5	1
		"วันพฤหัสบดี"	1
		6	1
		"วันศุกร์"	1
		7	1
		"วันเสาร์"	1
		"เลขไม่ถูกต้อง"	1
$n_1 = 11$	$N_1 = 33$	$n_2 = 17$	$N_2 = 18$

ตารางที่ 3.6 ตัวแปรตีแผ่เฉพาะถิ่น การกำหนดค่าตัวแปรครั้งก่อน และความซับซ้อนจากการไหลของข้อมูล ของโปรแกรมทดสอบที่ 2

โหนดที่	ตัวแปรตีแผ่เฉพาะถิ่น	การกำหนดค่าตัวแปรครั้งก่อน	df _i
2	I	I(1)	1
		df	1

ปัจจัยทั้งหมดที่ได้ของโปรแกรมทดสอบที่ 2 สรุปได้ดังนี้

$$n = 13, \quad e = cf = 19$$

$$n_1 = 11, \quad n_2 = 17$$

$$N_1 = 33, \quad N_2 = 18$$

$$df = 1$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำค่าที่ได้แทนในสูตรตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo จะได้ค่าความซับซ้อนดังแสดงในตารางที่ 3.7

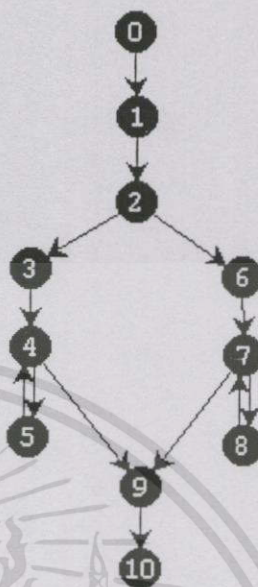
ตารางที่ 3.7 ค่าความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo ของโปรแกรมทดสอบที่ 2

ตัววัด	ความซับซ้อน
Halstead	$11 \times 18 \times (33 + 18) \log_2(11 + 17) / 2 \times 17 = 1,427.784$
McCabe	$19 - 13 + 2 = 8$
Oviedo	$19 + 1 = 20$

โปรแกรมทดสอบที่ 2 แสดงการเขียนโปรแกรมโครงสร้าง Select Case ซึ่งให้ผลลัพธ์การทำงานเหมือนกับโปรแกรมทดสอบที่ 1 ผลการทดสอบวัดความซับซ้อนของโปรแกรมทดสอบที่ 2 จากตารางที่ 3.7 ค่าความซับซ้อนจากตัววัดของ Halstead มีค่าเท่ากับ 1,427.784 ส่วนค่าความซับซ้อนจากตัววัดของ McCabe มีค่าเท่ากับ 8 และค่าความซับซ้อนจากตัววัดของ Oviedo เท่ากับ 20

โปรแกรมทดสอบที่ 3

Line	Node	Code
1	0	Private Sub lblCheck_Click(Index As Integer)
2	1	Dim i As Integer
3	2	Select Case Index
4	3	Case 1
5	3	lblMark(2).Visible = False
6	3	lblMark(1).Visible = True
7	4	For i = 1 To 3 Step 1
8	5	PanTime(i) = ArrayTime(i, 1)
9	5	Next
10	6	Case 2
11	6	lblMark(1).Visible = False
12	6	lblMark(2).Visible = True
13	7	For i = 1 To 3 Step 1
14	8	PanTime(i) = ArrayTime(i, 2)
15	8	Next
16	9	End Select
17	10	End Sub



รูปที่ 3.4

(ก) โปรแกรมทดสอบที่ 3

(ข) กราฟควบคุมการไหลของโปรแกรมทดสอบที่ 3

จากรูปที่ 3.4 (ก) แสดงโปรแกรมทดสอบที่ 3 สามารถกำหนดขอบเขตของโหนดได้ดังนี้ โปรแกรมบรรทัดที่ 1 เป็นสมาชิกในโหนดที่ 0 โปรแกรมบรรทัดที่ 2 เป็นสมาชิกในโหนดที่ 1 โปรแกรมบรรทัดที่ 3 เป็นสมาชิกในโหนดที่ 2 โปรแกรมบรรทัดที่ 4 ถึง 6 เป็นสมาชิกในโหนดที่ 3 โปรแกรมบรรทัดที่ 7 เป็นสมาชิกในโหนดที่ 4 โปรแกรมบรรทัดที่ 8 ถึง 9 เป็นสมาชิกในโหนดที่ 5 โปรแกรมบรรทัดที่ 10 ถึง 12 เป็นสมาชิกในโหนดที่ 6 โปรแกรมบรรทัดที่ 13 เป็นสมาชิกในโหนดที่ 7 โปรแกรมบรรทัดที่ 14 ถึง 15 เป็นสมาชิกในโหนดที่ 8 โปรแกรมบรรทัดที่ 16 เป็นสมาชิกในโหนดที่ 9 และโปรแกรมบรรทัดที่ 17 เป็นสมาชิกในโหนดที่ 10 ดังนั้นจึงสามารถสร้างกราฟควบคุมการไหลได้ดังรูปที่ 3.4 (ข) จำนวนโหนดที่ได้เท่ากับ 11 จำนวนเส้นเชื่อมที่ได้เท่ากับ 13 และสามารถแสดงจำนวนโอเปอเรเตอร์และโอเปอเรนด์ได้ดังตารางที่ 3.8 ส่วนตัวแปรที่ได้รับการกำหนดค่าและตัวแปรตีแผ่เฉพาะถิ่นแสดงได้ดังตารางที่ 3.9

ตารางที่ 3.8 จำนวนโอเปอเรเตอร์และโอเปอเรนด์ของโปรแกรมทดสอบที่ 3

โอเปอเรเตอร์	จำนวน	โอเปอเรนด์	จำนวน
Private	1	Index	2
Sub	1	1	8
As	1	lblMark().Visible	4
()	9	2	4
Integer	1	False	2
Select Case	1	True	2
Case	2	i	6
.	4	3	2
=	8	PanTime	2
For	2	ArrayTime	2
To	2		
Step	2		
,	2		
Next	2		
End Select	1		
End Sub	1		
$n_1 = 16$	$N_1 = 40$	$n_2 = 10$	$N_2 = 34$

ตารางที่ 3.9 ตัวแปรตีแผ่เฉพาะถิ่น การกำหนดค่าตัวแปรครั้งก่อน และความซับซ้อนจากการไหลของข้อมูล ของโปรแกรมทดสอบที่ 3

โหนดที่	ตัวแปรตีแผ่เฉพาะถิ่น	การกำหนดค่าตัวแปรครั้งก่อน	df _i
2	Index	Index(0)	1
5	i	i(4)	1
8	I	i(7)	1
		df	3

ปัจจัยทั้งหมดที่ได้ของโปรแกรมทดสอบที่ 3 สรุปได้ดังนี้

$$n = 11, \quad e = cf = 13$$

$$n_1 = 16, \quad n_2 = 10$$

$$N_1 = 40, \quad N_2 = 34$$

$$df = 3$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำค่าที่ได้แทนในสูตรตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo จะได้ค่าความซับซ้อนดังแสดงในตารางที่ 3.10

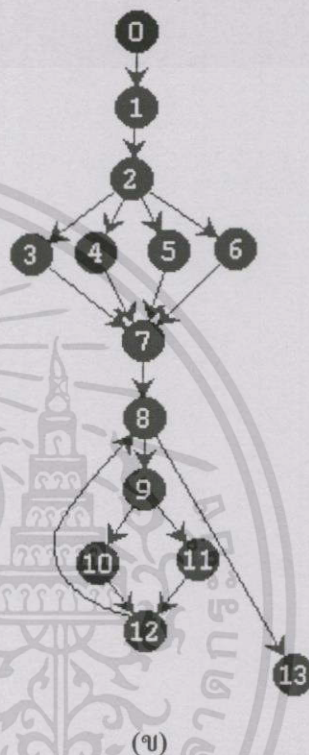
ตารางที่ 3.10 ค่าความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo ของโปรแกรมทดสอบที่ 3

ตัววัด	ความซับซ้อน
Halstead	$16 \times 34 \times (40 + 34) \log_2(16 + 10) / 2 \times 10 = 9,461.045$
McCabe	$13 - 11 + 2 = 4$
Oviedo	$13 + 3 = 16$

โปรแกรมทดสอบที่ 3 แสดงการเขียนโปรแกรมโครงสร้าง Select Case ซึ่งมีการซ้อนใน (Nested) ด้วนลูป (Loop) For ผลการทดสอบวัดความซับซ้อนของโปรแกรมทดสอบที่ 3 จากตารางที่ 3.10 ค่าความซับซ้อนจากตัววัดของ Halstead มีค่าเท่ากับ 9,461.058 ส่วนค่าความซับซ้อนจากตัววัดของ McCabe มีค่าเท่ากับ 4 และค่าความซับซ้อนจากตัววัดของ Oviedo เท่ากับ 16

โปรแกรมทดสอบที่ 4

Line	Node	Code
1	0	Sub tabMtxFile_RowChange ()
2	1	Dim i As Integer
3	2	Select Case tabMtxFile.ColumnText(11)
4	3	Case "INI"
5	3	cmdMtxFile(0).Enabled = True
6	3	cmdMtxFile(1).Enabled = True
7	3	cmdMtxFile(2).Enabled = True
8	4	Case "RUN"
9	4	cmdMtxFile(0).Enabled = False
10	4	cmdMtxFile(1).Enabled = True
11	4	cmdMtxFile(2).Enabled = False
12	5	Case "ABT"
13	5	cmdMtxFile(0).Enabled = True
14	5	cmdMtxFile(1).Enabled = False
15	5	cmdMtxFile(2).Enabled = True
16	6	Case "DEL"
17	6	cmdMtxFile(0).Enabled = False
18	6	cmdMtxFile(1).Enabled = True
19	6	cmdMtxFile(2).Enabled = False
20	7	End Select
21	8	For i = 0 To 2
22	9	If cmdMtxFile(i).Enabled Then
23	10	cmdMtxFile(i).ForeColor = &HFF00&
24	11	Else
25	11	cmdMtxFile(i).ForeColor = &HFF&
26	12	End If
27	12	Next i
28	13	End Sub



รูปที่ 3.5 (ก) โปรแกรมทดสอบที่ 4

(ข) กราฟควบคุมการไหลของโปรแกรมทดสอบที่ 4

จากรูปที่ 3.5 (ก) แสดงโปรแกรมทดสอบที่ 2 สามารถกำหนดขอบเขตของโหนดได้ดังนี้ โปรแกรมบรรทัดที่ 1 เป็นสมาชิกในโหนดที่ 0 โปรแกรมบรรทัดที่ 2 เป็นสมาชิกในโหนดที่ 1 โปรแกรมบรรทัดที่ 3 เป็นสมาชิกในโหนดที่ 2 โปรแกรมบรรทัดที่ 4 ถึง 7 เป็นสมาชิกในโหนดที่ 3 โปรแกรมบรรทัดที่ 8 ถึง 11 เป็นสมาชิกในโหนดที่ 4 โปรแกรมบรรทัดที่ 12 ถึง 15 เป็นสมาชิกในโหนดที่ 5 โปรแกรมบรรทัดที่ 16 ถึง 19 เป็นสมาชิกในโหนดที่ 6 โปรแกรมบรรทัดที่ 20 เป็นสมาชิกในโหนดที่ 7 โปรแกรมบรรทัดที่ 21 เป็นสมาชิกในโหนดที่ 8 โปรแกรมบรรทัดที่ 22 เป็นสมาชิกในโหนดที่ 9 โปรแกรมบรรทัดที่ 23 เป็นสมาชิกในโหนดที่ 10 โปรแกรมบรรทัดที่ 24 ถึง

25 เป็นสมาชิกในโหนดที่ 11 โปรแกรมบรรทัดที่ 26 ถึง 27 เป็นสมาชิกในโหนดที่ 12 และโปรแกรมบรรทัดที่ 28 เป็นสมาชิกในโหนดที่ 13 ดังนั้นจึงสามารถสร้างกราฟควบคุมการไหลได้ดังรูปที่ 3.5 (ข) จำนวนโหนดที่ได้เท่ากับ 14 จำนวนเส้นเชื่อมที่ได้เท่ากับ 18 และสามารถแสดงจำนวนโอเปอเรเตอร์และโอเปอเรนด์ได้ดังตารางที่ 3.11 ส่วนตัวแปรที่ได้รับการกำหนดค่าและตัวแปรพิเศษเฉพาะถิ่นแสดงได้ดังตารางที่ 3.12

ตารางที่ 3.11 จำนวน โอเปอเรเตอร์และโอเปอเรนด์ของโปรแกรมทดสอบที่ 4

โอเปอเรเตอร์	จำนวน	โอเปอเรนด์	จำนวน
Sub	1	TabMtxFile.ColumText()	1
()	17	!!	1
Select Case	1	"INI"	1
.	16	cmdMtxFile().Enabled	12
Case	4	0	5
""	4	1	4
=	15	2	5
End Select	1	True	7
For	1	"RUN"	1
To	1	False	5
If	1	"ABT"	1
Then	1	"DEL"	1
Else	1	I	5
End If	1	cmdMtxFile().Enabled	1
Next	1	cmdMtxFile().ForeColor	2
End Sub	1	&HFF00&	1
		&HFF&	1
$n_1 = 16$	$N_1 = 67$	$n_2 = 17$	$N_2 = 54$

ตารางที่ 3.12 ตัวแปรตีแผ่เฉพาะถิ่น การกำหนดค่าตัวแปรครั้งก่อน และความซับซ้อนจากการไหลของข้อมูล ของโปรแกรมทดสอบที่ 4

โนหนดที่	ตัวแปรตีแผ่เฉพาะถิ่น	การกำหนดค่าตัวแปรครั้งก่อน	df _i
9	i	i(8)	1
	cmdMtxFile(i).Enabled	cmdMtxFile(0).Enabled(3) , cmdMtxFile(1).Enabled(3) , cmdMtxFile(2).Enabled(3) , cmdMtxFile(0).Enabled(4) , cmdMtxFile(1).Enabled(4) , cmdMtxFile(2).Enabled(4) , cmdMtxFile(0).Enabled(5) , cmdMtxFile(1).Enabled(5) , cmdMtxFile(2).Enabled(5) , cmdMtxFile(0).Enabled(6) , cmdMtxFile(1).Enabled(6) , cmdMtxFile(2).Enabled(6)	12
12	i	i(8)	1
		df	14

ปัจจัยทั้งหมดที่ได้ของ โปรแกรมทดสอบที่ 4 สรุปได้ดังนี้

$$n = 14, \quad e = cf = 18$$

$$n_1 = 16, \quad n_2 = 17$$

$$N_1 = 67, \quad N_2 = 54$$

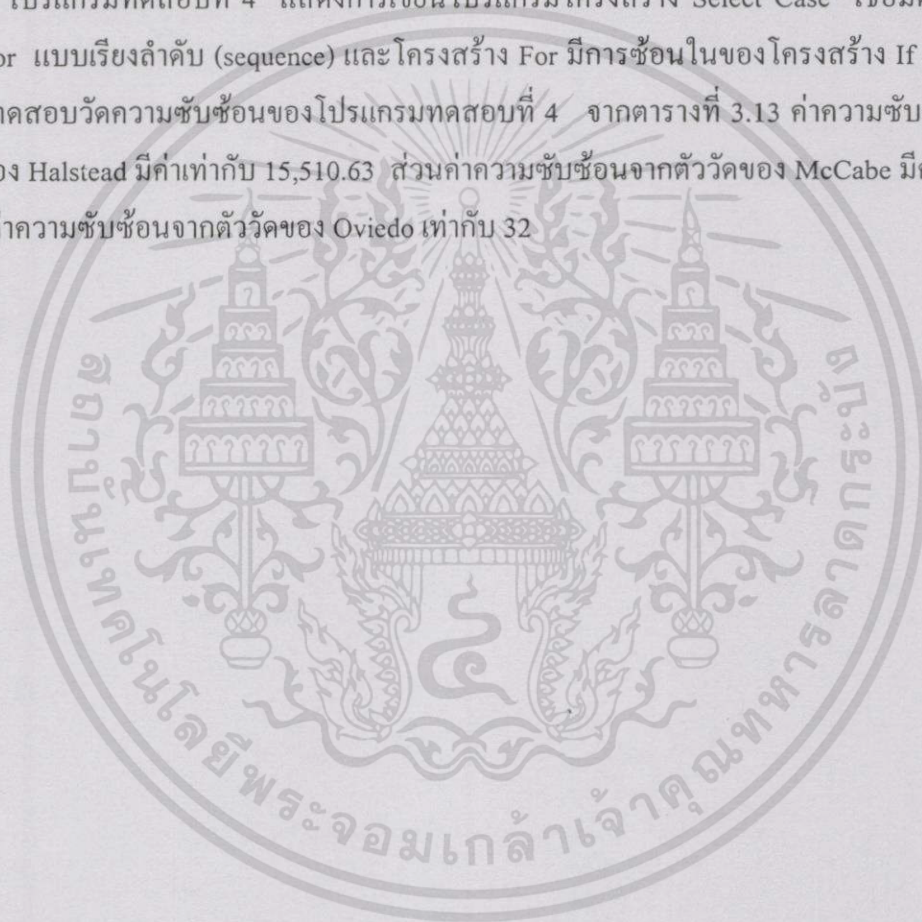
$$df = 14$$

เมื่อนำค่าที่ได้แทนในสูตรตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo จะได้ค่าความซับซ้อนดังแสดงในตารางที่ 3.13

ตารางที่ 3.13 ค่าความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo ของโปรแกรมทดสอบที่ 4

ตัววัด	ความซับซ้อน
Halstead	$16 \times 54 \times (67 + 54) \log_2(16 + 17) / 2 \times 17 = 15,510.630$
McCabe	$18 - 14 + 2 = 6$
Oviedo	$18 + 14 = 32$

โปรแกรมทดสอบที่ 4 แสดงการเขียนโปรแกรมโครงสร้าง Select Case เชื่อมด้วยโครงสร้าง For แบบเรียงลำดับ (sequence) และโครงสร้าง For มีการซ้อนในของโครงสร้าง If . . . Else ผลการทดสอบวัดความซับซ้อนของโปรแกรมทดสอบที่ 4 จากตารางที่ 3.13 ค่าความซับซ้อนจากตัววัดของ Halstead มีค่าเท่ากับ 15,510.63 ส่วนค่าความซับซ้อนจากตัววัดของ McCabe มีค่าเท่ากับ 6 และค่าความซับซ้อนจากตัววัดของ Oviedo เท่ากับ 32

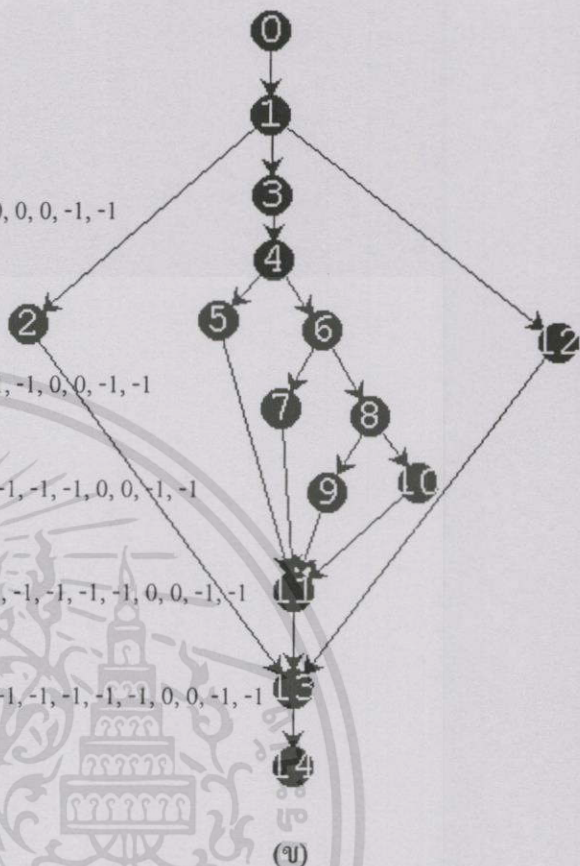


โปรแกรมทดสอบที่ 5

Line Node

```

1 0 Private Sub spiRollback()
2 1   Select Case mAddUp_flag
3 2     Case "A" ' กรณี Add
4 2       fpiFEnable (0)
5 2       fpbClearText Me
6         'spbInitToolBar Me, -1, 0, 0, 0, 0, -1, 0, 0, 0, 0, -1, -1
7 2       spbInitToolBar "Init"
8 3     Case "U" ' กรณี Update
9 4       If mRecordAmt = 1 Then
10 5         spbInitToolBar "One"
11         'spbInitToolBar Me, -1, 0, 0, 0, 0, -1, -1, -1, 0, 0, -1, -1
12 6       ElseIf mRecordNo = mRecordAmt Then
13 7         spbInitToolBar "Prev"
14         'spbInitToolBar Me, -1, -1, -1, 0, 0, -1, -1, -1, 0, 0, -1, -1
15 8       ElseIf mRecordNo = 1 Then
16 9         spbInitToolBar "Next"
17         'spbInitToolBar Me, -1, 0, 0, -1, -1, -1, -1, -1, 0, 0, -1, -1
18 10      Else
19 10        spbInitToolBar "Between"
20         'spbInitToolBar Me, -1, -1, -1, -1, -1, -1, -1, 0, 0, -1, -1
21 11      End If
22 11      spiShowData
23 11      fpiFEnable (0)
24 11      gQryDyn.Update
25 12     Case "Q"
26 12       spiEndQuery
27 13     End Select
28 14   End Sub
    
```



รูปที่ 3.6 (ก) โปรแกรมทดสอบที่ 5

(ข) กราฟควบคุมการไหลของโปรแกรมทดสอบที่ 5

จากรูปที่ 3.6 (ก) แสดงโปรแกรมทดสอบที่ 5 สามารถกำหนดขอบเขตของโหนดได้ดังนี้ โปรแกรมบรรทัดที่ 1 เป็นสมาชิกในโหนดที่ 0 โปรแกรมบรรทัดที่ 2 เป็นสมาชิกในโหนดที่ 1 โปรแกรมบรรทัดที่ 3 ถึง 6 เป็นสมาชิกในโหนดที่ 2 โปรแกรมบรรทัดที่ 7 เป็นสมาชิกในโหนดที่ 3 โปรแกรมบรรทัดที่ 8 เป็นสมาชิกในโหนดที่ 4 โปรแกรมบรรทัดที่ 9 เป็นสมาชิกในโหนดที่ 5 โปรแกรมบรรทัดที่ 10 เป็นสมาชิกในโหนดที่ 6 โปรแกรมบรรทัดที่ 11 เป็นสมาชิกในโหนดที่ 7 โปรแกรมบรรทัดที่ 12 เป็นสมาชิกในโหนดที่ 8 โปรแกรมบรรทัดที่ 13 เป็นสมาชิกในโหนดที่ 9 โปรแกรมบรรทัดที่ 14 ถึง 15 เป็นสมาชิกในโหนดที่ 10 โปรแกรมบรรทัดที่ 16 ถึง 19 เป็นสมาชิกในโหนดที่ 11 โปรแกรมบรรทัดที่ 20 ถึง 21 เป็นสมาชิกในโหนดที่ 12 โปรแกรมบรรทัดที่ 22 เป็นสมาชิกในโหนดที่ 13 และ โปรแกรมบรรทัดที่ 23 เป็นสมาชิกในโหนดที่ 14 ดังนั้นจึงสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างกราฟควบคุมการไหลได้ดังรูปที่ 3.6 (ข) จำนวนโหนดที่ได้เท่ากับ 15 จำนวนเส้นเชื่อมที่ได้เท่ากับ 19 และสามารถแสดงจำนวนโอเปอเรเตอร์และโอเปอเรนด์ได้ดังตารางที่ 3.14 ในโพธิ์เจอร์นี้ไม่พบตัวแปรตัวแปรพิเศษเฉพาะถิ่น ดังนั้นค่า df จึงเท่ากับ 0

ตารางที่ 3.14 จำนวน โอเปอเรเตอร์และโอเปอเรนด์ของโปรแกรมทดสอบที่ 5

โอเปอเรเตอร์	จำนวน	โอเปอเรนด์	จำนวน
Private	1	mAddUp_flag	1
Sub	1	"A"	1
()	3	0	2
Select Case	1	Me	1
Case	3	"Init"	1
""	8	"U"	1
fpiFEnable	2	mRecordAmt	2
fpbClearText	1	1	2
spbInitToolBar	5	"One"	1
If	1	mRecordNo	2
=	3	"Prev"	1
Then	3	"Next"	1
ElseIf	2	"Between"	1
Else	1	gQryDyn	1
End If	1	"Q"	1
.	1		
End Select	1		
End Sub	1		
$n_1 = 18$	$N_1 = 39$	$n_2 = 15$	$N_2 = 19$

ปัจจัยทั้งหมดที่ได้ของโปรแกรมทดสอบที่ 5 สรุปได้ดังนี้

$$\begin{aligned}
 n &= 15, & e &= cf = 19 \\
 n_1 &= 18, & n_2 &= 15 \\
 N_1 &= 39, & N_2 &= 19 \\
 df &= 0
 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำค่าที่ได้แทนในสูตรตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo จะได้ค่าความซับซ้อนดังแสดงในตารางที่ 3.15

ตารางที่ 3.15 ค่าความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo ของโปรแกรมทดสอบที่ 5

ตัววัด	ความซับซ้อน
Halstead	$18 \times 19 \times (39 + 19) \log_2(18 + 15) / 2 \times 15 = 3,335.355$
McCabe	$19 - 15 + 2 = 6$
Oviedo	$19 + 0 = 19$

โปรแกรมทดสอบที่ 5 แสดงการเขียนโปรแกรมโครงสร้าง Select Case ซึ่งมีการซ้อนในของโครงสร้าง If... ElseIf ผลการทดสอบวัดความซับซ้อนของโปรแกรมทดสอบที่ 5 จากตารางที่ 3.15 ค่าความซับซ้อนจากตัววัดของ Halstead มีค่าเท่ากับ 3,335.355 ส่วนค่าความซับซ้อนจากตัววัดของ McCabe มีค่าเท่ากับ 6 และค่าความซับซ้อนจากตัววัดของ Oviedo เท่ากับ 19

โปรแกรมทดสอบที่ 6

```

Line Node
1 0 Sub Command1_Click()
2 1 Dim a(3,2) As Integer
3 1 Dim b(2,3) As Integer
4 1 Dim c(3,3) As Integer
5 1 Dim n, m, i, j, k As Integer

6 2 For i = 0 To 3
7 3 For j = 0 To 3
8 4 c(i,j) = 0
9 4 Next j
10 5 Next i
11 6 k = 0
12 7 For i = 0 To 2
13 8 For j = 0 To 1
14 9 a(i,j) = Val(Text(k))
15 9 k = k + 1
16 9 Next j
17 10 Next i
18 11 k = 0
19 12 For i = 0 To 1
20 13 For j = 0 To 2
21 14 b(i,j) = Val(Text(k))
22 14 k = k + 1
23 14 Next j
24 15 Next i
25 16 For i = 0 To n - 1
26 17 For j = 0 To n - 1
27 18 For k = 0 To m - 1
28 19 c(i,j) = c(i,j) + (a(i,k) * b(k,j))
29 19 Next k
30 20 Print c(i,j)
31 21 Next j
32 22 Next i
33 23 End Sub

```



(ก)

(ข)

รูปที่ 3.7 (ก) โปรแกรมทดสอบที่ 6

(ข) กราฟควบคุมการไหลของโปรแกรมทดสอบที่ 6

จากรูปที่ 3.7 (ก) แสดงโปรแกรมทดสอบที่ 6 สามารถกำหนดขอบเขตของโหนดได้ดังนี้ โปรแกรมบรรทัดที่ 1 เป็นสมาชิกในโหนดที่ 0 โปรแกรมบรรทัดที่ 2 ถึง 5 เป็นสมาชิกในโหนดที่ 1 โปรแกรมบรรทัดที่ 6 เป็นสมาชิกในโหนดที่ 2 โปรแกรมบรรทัดที่ 7 เป็นสมาชิกในโหนดที่ 3 โปรแกรมบรรทัดที่ 8 ถึง 9 เป็นสมาชิกในโหนดที่ 4 โปรแกรมบรรทัดที่ 10 เป็นสมาชิกในโหนดที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 โปรแกรมบรรทัดที่ 11 เป็นสมาชิกในโหนดที่ 6 โปรแกรมบรรทัดที่ 12 เป็นสมาชิกในโหนดที่ 7 โปรแกรมบรรทัดที่ 13 เป็นสมาชิกในโหนดที่ 8 โปรแกรมบรรทัดที่ 14 ถึง 16 เป็นสมาชิกในโหนดที่ 9 โปรแกรมบรรทัดที่ 17 เป็นสมาชิกในโหนดที่ 10 โปรแกรมบรรทัดที่ 18 เป็นสมาชิกในโหนดที่ 11 โปรแกรมบรรทัดที่ 19 เป็นสมาชิกในโหนดที่ 12 โปรแกรมบรรทัดที่ 20 เป็นสมาชิกในโหนดที่ 13 โปรแกรมบรรทัดที่ 21 ถึง 23 เป็นสมาชิกในโหนดที่ 14 โปรแกรมบรรทัดที่ 24 เป็นสมาชิกในโหนดที่ 15 โปรแกรมบรรทัดที่ 25 เป็นสมาชิกในโหนดที่ 16 โปรแกรมบรรทัดที่ 26 เป็นสมาชิกในโหนดที่ 17 โปรแกรมบรรทัดที่ 27 เป็นสมาชิกในโหนดที่ 18 โปรแกรมบรรทัดที่ 28 ถึง 29 เป็นสมาชิกในโหนดที่ 19 โปรแกรมบรรทัดที่ 30 เป็นสมาชิกในโหนดที่ 20 โปรแกรมบรรทัดที่ 31 เป็นสมาชิกในโหนดที่ 21 โปรแกรมบรรทัดที่ 32 เป็นสมาชิกในโหนดที่ 22 และโปรแกรมบรรทัดที่ 33 เป็นสมาชิกในโหนดที่ 23 ดังนั้นจึงสามารถสร้างกราฟควบคุมการไหลได้ดังรูปที่ 3.7 (ข) จำนวนโหนดที่ได้เท่ากับ 24 จำนวนเส้นเชื่อมที่ได้เท่ากับ 32 และสามารถแสดงจำนวนโอเปอเรเตอร์และโอเปอเรนด์ได้ดังตารางที่ 3.15 ส่วนตัวแปรที่ได้รับการกำหนดค่าและตัวแปรตีแผ่เฉพาะถิ่นแสดงได้ดังตารางที่ 3.16

ตารางที่ 3.16 จำนวนโอเปอเรเตอร์และโอเปอเรนด์ของโปรแกรมทดสอบที่ 6

โอเปอเรเตอร์	จำนวน	โอเปอเรนด์	จำนวน
Sub	1	i	16
()	16	0	12
For	9	3	2
=	17	j	16
To	9	c	4
,	9	k	12
Next	9	2	2
Val	2	l	7
+	4	a	3
-	2	Text	2
*	1	b	2
Print	1	n	2
End Sub	1	m	1
$n_1 = 13$	$N_1 = 82$	$n_2 = 13$	$N_2 = 81$

ตารางที่ 3.17 ตัวแปรตีแผ่เฉพาะถิ่น การกำหนดค่าตัวแปรครั้งก่อน และความซับซ้อนจากการไหลของข้อมูล ของโปรแกรมทดสอบที่ 6

โหนดที่	ตัวแปรตีแผ่เฉพาะถิ่น	การกำหนดค่าตัวแปรครั้งก่อน	df _i
4	i	i(2)	1
	j	i(3)	1
5	i	i(2)	1
9	i	i(2), i(7)	2
	j	j(3), j(8)	2
	k	k(6)	1
10	i	i(2), i(7)	2
14	i	i(2), i(7), i(12)	3
	j	j(3), j(8), j(13)	3
	k	k(6), k(9), k(11)	3
15	i	i(2), i(7), i(12)	3
19	c	c(4)	1
	a	a(9)	1
	b	b(14)	1
	i	i(2), i(7), i(12), i(16)	4
	j	j(3), j(8), j(13), j(17)	4
	k	k(6), k(9), k(11), k(14), k(18)	5
20	c	c(4), c(19)	2
	i	i(2), i(7), i(12), i(16)	4
	j	j(3), j(8), j(13), j(17)	4
21	j	j(3), j(8), j(13), j(17)	4
22	i	i(2), i(7), i(12), i(16)	4
df			56

ปัจจัยทั้งหมดที่ได้ของโปรแกรมทดสอบที่ 5 สรุปได้ดังนี้

$$n = 24 , \quad e = cf = 32$$

$$n_1 = 13 , \quad n_2 = 13$$

$$N_1 = 82 , \quad N_2 = 81$$

$$df = 56$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำค่าที่ได้แทนในสูตรตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo จะได้ค่าความซับซ้อนดังแสดงในตารางที่ 3.17

ตารางที่ 3.18 ค่าความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo ของโปรแกรมทดสอบที่ 6

ตัววัด	ความซับซ้อน
Halstead	$12 \times 81 \times (82 + 81) \log_2(13 + 13) / 2 \times 13 = 31,029.953$
McCabe	$32 - 24 + 2 = 10$
Oviedo	$32 + 56 = 88$

โปรแกรมทดสอบที่ 6 แสดงการเขียนโปรแกรมการคำนวณทางคณิตศาสตร์ คือ โปรแกรมการคูณเมตริกซ์ขนาด $n \times m$ ผลการทดสอบวัดความซับซ้อนของโปรแกรมทดสอบที่ 6 จากตารางที่ 3.17 ค่าความซับซ้อนจากตัววัดของ Halstead มีค่าเท่ากับ 31,029.953 ส่วนค่าความซับซ้อนจากตัววัดของ McCabe มีค่าเท่ากับ 10 และค่าความซับซ้อนจากตัววัดของ Oviedo เท่ากับ 88

จากการทดสอบวัดความซับซ้อนของโปรแกรมทั้ง 6 โปรแกรม พบว่าค่าความซับซ้อนจากตัววัดของ Halstead จะให้ค่าตัวเลขที่สูงมากและตัวเลขที่ได้ให้ค่าเป็นทศนิยม เนื่องจากมีสูตรการคำนวณที่ซับซ้อน ส่วนค่าความซับซ้อนจากตัววัดของ McCabe และ Oviedo จะเป็นจำนวนเต็ม ตัวเลขของแต่ละโปรแกรมที่ได้จากตัววัดของ McCabe จะให้ค่าต่ำสุด ค่าความซับซ้อนจากตัววัดของ Oviedo จะสูงกว่าเนื่องจากตัววัดของ McCabe พิจารณาเฉพาะโครงสร้างควบคุมการไหลเท่านั้น ตัววัดของ Oviedo จะพิจารณาทั้งโครงสร้างควบคุมการไหลและการไหลของข้อมูล

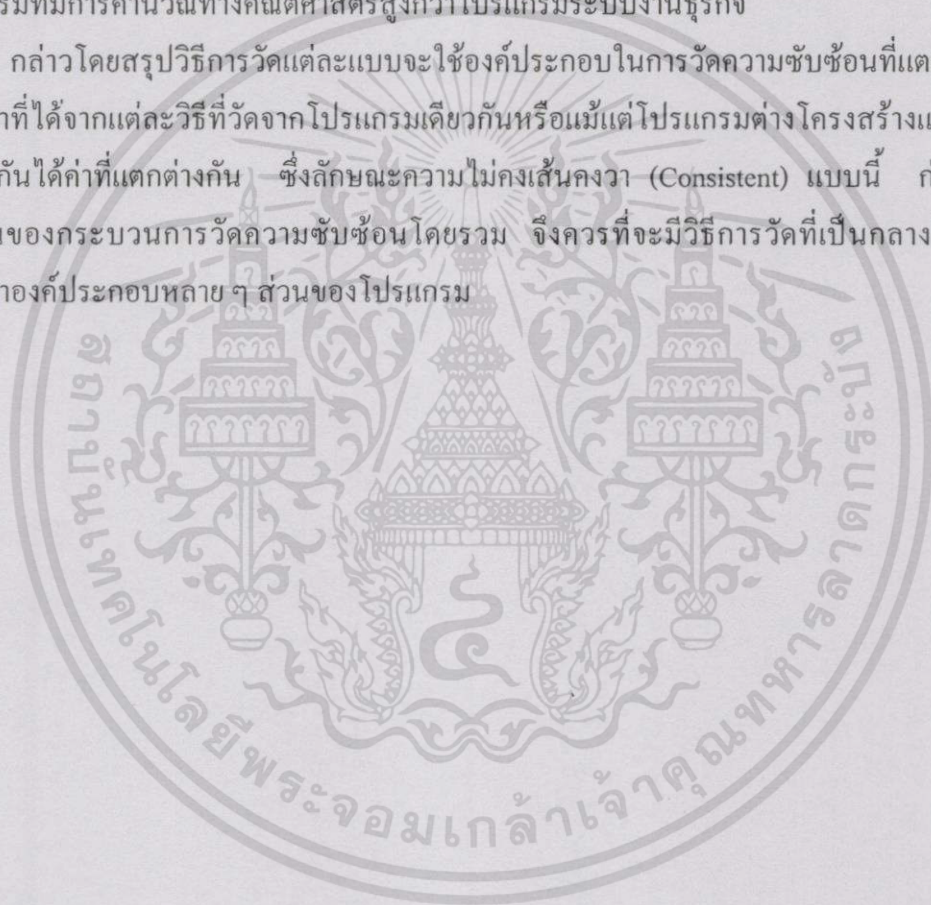
เมื่อเปรียบเทียบค่าความซับซ้อนระหว่างโปรแกรมทดสอบที่ 1 กับโปรแกรมทดสอบที่ 2 ซึ่งมีผลลัพธ์การทำงานเหมือนกันแต่ใช้โครงสร้างการเขียนโปรแกรมไม่เหมือนกัน ค่าความซับซ้อนจากตัววัดของ Halstead และตัววัดของ Oviedo เป็นแนวทางเดียวกันคือโปรแกรมทดสอบที่ 2 ให้ค่าความซับซ้อนน้อยกว่าโปรแกรมทดสอบที่ 1 แต่ตัววัดของ McCabe ให้ค่าความซับซ้อนของทั้ง 2 โปรแกรมเท่ากัน และในทำนองเดียวโปรแกรมทดสอบที่ 4 กับโปรแกรมทดสอบที่ 5 ซึ่งมีผลลัพธ์การทำงานต่างกัน ค่าความซับซ้อนจากตัววัดของ Halstead และตัววัดของ Oviedo เป็นแนวทางเดียวกันคือ โปรแกรมทดสอบที่ 4 ให้ค่าความซับซ้อนมากกว่าโปรแกรมทดสอบที่ 5 ส่วนตัววัดของ McCabe ให้ค่าความซับซ้อนเท่ากัน จากผลการวัดนี้จะพบว่าตัววัดของ McCabe มีจุดอ่อนในการวัดกรณีที่ใช้การเขียนโปรแกรมโครงสร้าง Select Case และจุดอ่อนนี้เป็นแนวทางในการนำเสนอตัววัดแบบใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเปรียบเทียบค่าความซับซ้อนระหว่างโปรแกรมทดสอบที่ 2 กับโปรแกรมทดสอบที่ 3 ซึ่งมีการทำงานแตกต่างกัน พบว่าตัววัดของ Halstead ให้ค่าความซับซ้อนของโปรแกรมทดสอบที่ 2 น้อยกว่าโปรแกรมทดสอบที่ 3 แต่ตัววัดของ McCabe และ Oviedo ให้ค่าความซับซ้อนในแนวทางเดียวกันคือโปรแกรมทดสอบที่ 2 มีค่าความซับซ้อนมากกว่าโปรแกรมทดสอบที่ 3 เนื่องจากตัววัดของ Halstead จะพิจารณาเฉพาะปริมาณของโอเปอเรเตอร์และโอเปอเรนด์เท่านั้น ไม่พิจารณาโครงสร้างของโปรแกรม

จากโปรแกรมทดสอบ 5 โปรแกรมแรกเป็นโปรแกรมระบบงานทางธุรกิจ ส่วนโปรแกรมทดสอบที่ 6 เป็นโปรแกรมที่มีการคำนวณทางคณิตศาสตร์ ซึ่งค่าความซับซ้อนของโปรแกรมที่มีการคำนวณทางคณิตศาสตร์สูงกว่าโปรแกรมระบบงานธุรกิจ

กล่าวโดยสรุปวิธีการวัดแต่ละแบบจะใช้องค์ประกอบในการวัดความซับซ้อนที่แตกต่างกัน ทำให้ค่าที่ได้จากแต่ละวิธีที่วัดจากโปรแกรมเดียวกันหรือแม้แต่โปรแกรมต่างโครงสร้างแต่ทำงานเหมือนกันได้ค่าที่แตกต่างกัน ซึ่งลักษณะความไม่คงเส้นคงวา (Consistent) แบบนี้ ก่อให้เกิดจุดอ่อนของกระบวนการวัดความซับซ้อนโดยรวม จึงควรที่จะมีวิธีการวัดที่เป็นกลางโดยการพิจารณาองค์ประกอบหลายๆ ส่วนของโปรแกรม



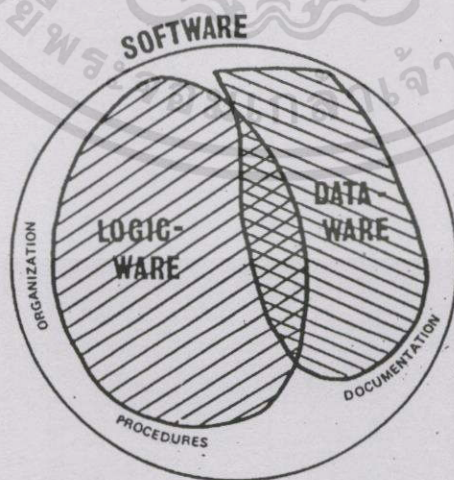
บทที่ 4

การออกแบบตัววัดความซับซ้อนของซอฟต์แวร์จาก โครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล

4.1 ความเป็นมาเกี่ยวกับตัววัดโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล

ในบทนี้เป็นการนำเสนอตัววัดความซับซ้อนของซอฟต์แวร์ ซึ่งแนวทางของผู้วิจัยเกิดจากตัววัดของ McCabe เนื่องจากตัววัดของ McCabe พิจารณาเฉพาะโครงสร้างควบคุมการไหล ผู้วิจัยคิดว่าควรมีปัจจัยอื่นเข้ามาเกี่ยวข้องเมื่อพิจารณาจากโปรแกรมทดสอบที่ 1 และโปรแกรมทดสอบที่ 2 ในบทที่ 3 เป็นโปรแกรมที่ให้ผลลัพธ์การทำงานเหมือนกันแต่ใช้เงื่อนไขการเขียนโปรแกรมต่างกัน ให้ค่าความซับซ้อนจากโครงสร้างของ McCabe หรือ $v(F)$ เท่ากัน ซึ่งตามหลักการเขียนโปรแกรมแบบโครงสร้าง การใช้เงื่อนไข Select Case นั้น จะเขียนโปรแกรมได้ง่ายกว่าและทำความเข้าใจได้ง่ายกว่าด้วย [16] ดังนั้นค่าความซับซ้อนของโปรแกรมทดสอบที่ 2 ควรจะน้อยกว่าโปรแกรมทดสอบที่ 1

ผู้วิจัยพบว่าปัจจัยที่ทำให้ความซับซ้อนจากโครงสร้างของโปรแกรมเพิ่มขึ้นก็คือจำนวนตัวแปรที่ถูกใช้ตรวจสอบภายใต้เงื่อนไข If . . . EndIf For . . . Next Do . . . Loop และ Select Case . . . End Select ซึ่งเป็นข้อมูลควบคุมการไหล ผู้วิจัยจึงได้นำเสนอตัววัดความซับซ้อนของโปรแกรมจากโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล เป็นคุณสมบัติที่ได้จากองค์ประกอบหลักของซอฟต์แวร์ Gilb [17] แบ่งองค์ประกอบหลักของซอฟต์แวร์เป็น 2 ส่วน ได้แก่ ลอจิกแวร์ (Logicware) และดาต้าแวร์ (Dataware) ดังแสดงในรูปที่ 4.1



รูปที่ 4.1 องค์ประกอบหลักของซอฟต์แวร์

(ที่มา: หนังสือ Software Metrics. 1977)

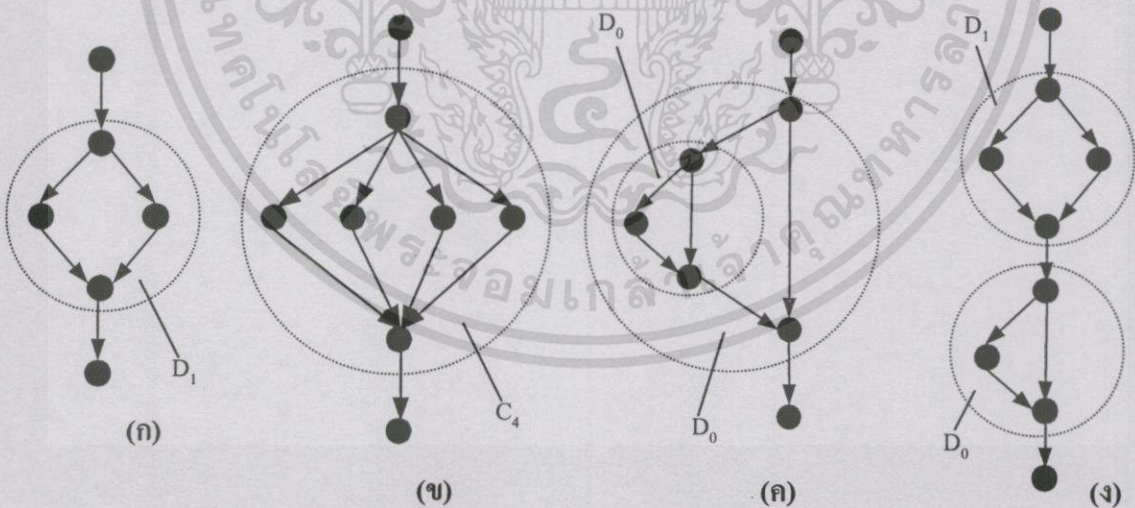
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นตัววัดความซับซ้อนจากโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล จึงเป็นวิธีการที่พิจารณาค่าความซับซ้อนจากองค์ประกอบทั้งสองส่วนของซอฟต์แวร์ ไม่ใช่ส่วนใดส่วนหนึ่งเพียงอย่างเดียว ซึ่งผู้วิจัยจะกล่าวถึงวิธีการวัดของตัววัดความซับซ้อนจากโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหลในหัวข้อต่อไป

4.2 ความซับซ้อนจากโครงสร้างควบคุมการไหล

ความซับซ้อนจากโครงสร้างควบคุมการไหล (Control Flow Structure Complexity , CFS) คือ จำนวนกราฟควบคุมการไหลย่อยที่เป็นกราฟควบคุมการไหลเฉพาะ ดังรูปที่ 2.5

ตัวอย่างการวัดค่าความซับซ้อนของโครงสร้างควบคุมการไหล พิจารณาจากรูปที่ 4.2 (ก) มีกราฟควบคุมการไหลเฉพาะอยู่ 2 โครงสร้าง เป็นโครงสร้างกราฟควบคุมการไหลเฉพาะ D_1 และ P ค่า CFS ที่ได้จะเท่ากับ 2 และในทำนองเดียวกันจากรูปที่ 4.2 (ข) ประกอบด้วยโครงสร้างกราฟควบคุมการไหลเฉพาะ C_4 และ P ค่า CFS ที่ได้มีค่าเท่ากับ 2 ส่วนรูปที่ 4.2 (ค) ค่า CFS จะได้เท่ากับ 3 เนื่องจากมีโครงสร้างกราฟควบคุมการไหลเฉพาะ D_0 อยู่ 2 โครงสร้าง และ P 1 โครงสร้าง ซึ่งซ้อนใน (nesting) กันอยู่ และรูปที่ 4.2 (ง) ค่า CFS ได้เท่ากับ 3 เนื่องจากมีโครงสร้างกราฟควบคุมการไหลเฉพาะ D_1 1 โครงสร้าง D_0 1 โครงสร้าง และ P 1 โครงสร้าง จะพบว่าทุก ๆ กราฟควบคุมการไหลจะต้องประกอบไปด้วยกราฟควบคุมการไหล P อยู่หนึ่งโครงสร้างเสมอ



- รูปที่ 4.2 (ก) โครงสร้างโปรแกรมที่ประกอบด้วยกราฟควบคุมการไหลเฉพาะ D_1
 (ข) โครงสร้างโปรแกรมที่ประกอบด้วยกราฟควบคุมการไหลเฉพาะ C_4
 (ค) โครงสร้างโปรแกรมที่ประกอบด้วยกราฟควบคุมการไหลเฉพาะ D_0 ซึ่งซ้อนในกัน
 (ง) โครงสร้างโปรแกรมที่ประกอบด้วยกราฟควบคุมการไหลเฉพาะ D_1 เรียงลำดับ D_0

4.3 ความซับซ้อนจากข้อมูลควบคุมการไหล

ความซับซ้อนจากข้อมูลควบคุมการไหล (Control Flow Data Complexity, CFD) คือ จำนวนตัวแปรหลักที่ใช้ตรวจสอบภายใต้เงื่อนไข รวมถึงฟังก์ชันที่มีการเรียกใช้ภายใต้เงื่อนไขด้วย ตัวอย่างเช่น โปรแกรมทดสอบที่ 1 และโปรแกรมทดสอบที่ 2 ในบทที่ 3 เมื่อพิจารณาปัจจัยความซับซ้อนของโปรแกรมทั้งสอง ปัจจัยที่ทำให้ความซับซ้อนเพิ่มขึ้นคือจำนวนตัวแปรที่ใช้ตรวจสอบเงื่อนไข ได้แก่ ตัวแปร I ซึ่งโปรแกรมทดสอบที่ 1 จำนวนตัวแปร I ถูกใช้ในการตรวจสอบเงื่อนไขทั้งหมด 7 ครั้ง ค่า CFD ได้เท่ากับ 7 แต่โปรแกรมทดสอบที่ 2 จำนวนตัวแปร I ถูกใช้ในการตรวจสอบเงื่อนไขเพียง 1 ครั้งเท่านั้น ค่า CFD จึงได้เท่ากับ 1

4.4 ตัววัดโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล

(Control Flow Structure and Control Flow Data Metric, SDM)

Card และ Glass [18] ได้กำหนดความซับซ้อนของซอฟต์แวร์ เท่ากับผลรวมความซับซ้อนของโครงสร้างกับความซับซ้อนของข้อมูล ซึ่งแสดงได้ดังนี้

$$\text{Program Complexity} = \text{Structural complexity} + \text{Data complexity} \quad (4.1)$$

จากสมการที่ 4.1 ผู้วิจัยกำหนดให้

$$\text{Structural complexity} = \text{CFS} \quad (4.2)$$

โดยที่ CFS แทนค่าความซับซ้อนจากโครงสร้างควบคุมการไหล

และกำหนดให้

$$\text{Data complexity} = \text{CFD} \quad (4.3)$$

โดยที่ CFD แทนค่าความซับซ้อนจากข้อมูลควบคุมการไหล

ดังนั้นค่าความซับซ้อนของตัววัด SDM สามารถคำนวณได้จากสูตร

$$\text{SDM} = \text{CFS} + \text{CFD} \quad (4.4)$$

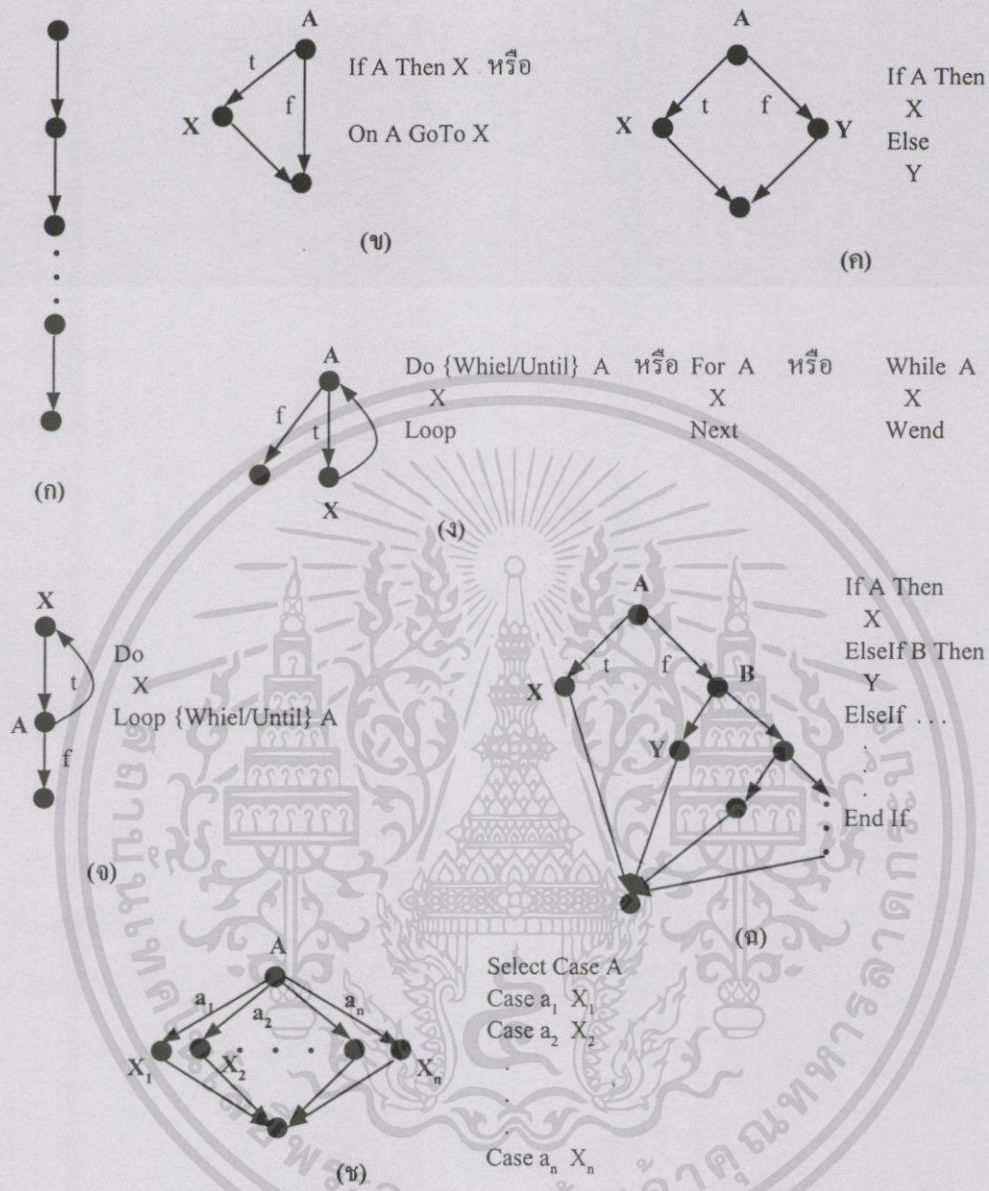
แต่เนื่องจาก CFS ของทุกโปรแกรมจะประกอบด้วยกราฟควบคุมการไหลเฉพาะ P อย่างน้อย 1 โครงสร้าง ดังนั้นผู้วิจัยจึงได้กำหนดค่าความซับซ้อนได้ดังนี้

$$SDM = CFS + CFD + 1 \quad (4.5)$$

โดยที่ CFS คือ จำนวนกราฟควบคุมการไหลเฉพาะที่ไม่ใช่กราฟควบคุมการไหล P
CFD คือ จำนวนตัวแปรหลักที่ใช้ตรวจสอบภายใต้เงื่อนไข

4.5 ปัจจัยของตัววัด SDM

เมื่อพิจารณาโครงสร้างของภาษาวิซวลเบสิกสามารถแสดงกราฟควบคุมการไหลเฉพาะตามโครงสร้างภาษาวิซวลเบสิกได้ดังรูปที่ 4.3 จากรูปที่ 4.3 (ก) เป็นกราฟควบคุมการไหลเฉพาะ P แสดงโครงสร้างควบคุมการไหลแบบเรียงลำดับซึ่งทุก ๆ กราฟควบคุมการไหลจะต้องประกอบด้วยกราฟควบคุมการไหล P อยู่ 1 โครงสร้าง รูปที่ 4.3 (ข) ถึง รูปที่ 4.3 (ฉ) แสดงโครงสร้างควบคุมการไหลของประโยคเงื่อนไข โดยให้ A แทนเงื่อนไข X แทนกลุ่มประโยคคำสั่งที่ต้องกระทำเมื่อเงื่อนไข A เป็นจริง รูปที่ 4.3 (ข) แสดงกราฟควบคุมการไหลเฉพาะ D_0 เป็นโครงสร้างควบคุมการไหลของประโยคเงื่อนไข If A Then X หรือ On A GoTo X รูปที่ 4.3 (ค) แสดงกราฟควบคุมการไหลเฉพาะ D_1 เป็นโครงสร้างควบคุมการไหลของประโยคเงื่อนไข If A Then X Else Y รูปที่ 4.3 (ง) แสดงกราฟควบคุมการไหลเฉพาะ D_2 เป็นโครงสร้างควบคุมการไหลของประโยคเงื่อนไข Do {While/Until} A X Loop หรือ For A X Next หรือ While A X Wend รูปที่ 4.3 (จ) แสดงกราฟควบคุมการไหลเฉพาะ D_3 เป็นโครงสร้างควบคุมการไหลของประโยคเงื่อนไข Do X Loop {While/Until} A รูปที่ 4.3 (ฉ) แสดงกราฟควบคุมการไหลเฉพาะ D_4 เป็นโครงสร้างควบคุมการไหลของประโยคเงื่อนไข If A Then X ElseIf B Then Y ElseIf ... End If รูปที่ 4.3 (ช) แสดงกราฟควบคุมการไหลเฉพาะ C เป็นโครงสร้างควบคุมการไหลของประโยคเงื่อนไข Select Case A Case a_1 X_1 ... Case a_n X_n



รูปที่ 4.3 (ก) กราฟควบคุมการไหลเฉพาะ P (ข) กราฟควบคุมการไหลเฉพาะ D₀
 (ค) กราฟควบคุมการไหลเฉพาะ D₁ (ง) กราฟควบคุมการไหลเฉพาะ D₂
 (จ) กราฟควบคุมการไหลเฉพาะ D₃ (ฉ) กราฟควบคุมการไหลเฉพาะ D₄
 (ช) กราฟควบคุมการไหลเฉพาะ C

ดังนั้นเราสามารถหาค่า CFS โดยพิจารณาจำนวนประโยคคำสั่งประเภทตัดสินใจ ได้แก่ คำสั่ง If , For , Do {While/Until} , While , Loop {While/Until} , Select Case และ On...GoTo จำนวน CFD พิจารณาจากจำนวนตัวแปรหลักที่นำมาใช้ควบคุมการไหลใน ประโยคคำสั่งประเภทตัดสินใจที่กล่าวมาข้างต้น รวมถึงฟังก์ชันที่มีการเรียกใช้ภายใต้เงื่อนไขด้วย เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

If $A = B$ And $C = D$

ตัวแปรหลักที่ควบคุมการไหลได้แก่ ตัวแปร A และ C ดังนั้น $CFD = 2$

Do While (A Or B) And C

ตัวแปรหลักที่ควบคุมการไหลได้แก่ ตัวแปร A B และ C ดังนั้น $CFD = 3$

If IsDate(A) Then

ตัวแปรหลักที่ควบคุมการไหล ได้แก่ ตัวแปร A และฟังก์ชันควบคุมการไหล ได้แก่ IsDate
ดังนั้น $CFD = 2$

4.6 การทดสอบวัดค่าความซับซ้อนจากตัววัด SDM

จากการวิเคราะห์ปัจจัยตามโครงสร้างภาษาวิซวลเบสิกของตัววัด SDM ผู้วิจัยจึงได้ทำการทดสอบโปรแกรมเพื่อหาค่าของปัจจัยที่เกี่ยวกับตัววัดนี้ โดยทำการทดสอบโปรแกรมทั้งหมด 5 โปรแกรม (โปรแกรมทดสอบทั้ง 5 โปรแกรมเป็นโปรแกรมเดียวกับโปรแกรมทดสอบในบทที่ 3) ดังต่อไปนี้

โปรแกรมทดสอบที่ 1 เมื่อพิจารณาโครงสร้างเฉพาะมีเพียงหนึ่งโครงสร้างเป็นโครงสร้างควบคุมการไหลของประโยคเงื่อนไข If... Elseif หรือ กราฟควบคุมการไหลเฉพาะ D_4 ดังนั้นค่า CFS เท่ากับ 1 และจากตารางที่ 4.1 ได้ค่า CFD เท่ากับ 7 เมื่อแทนค่าในสูตร $SDM = CFS + CFD + 1$ จึงได้ค่าความซับซ้อนเท่ากับ $1 + 7 + 1 = 9$

ตารางที่ 4.1 จำนวนข้อมูลควบคุมการไหลในโนหนดเงื่อนไขของโปรแกรมทดสอบที่ 1

โนหนด	ข้อมูลควบคุมการไหล	CFD_i
2	I	1
4	I	1
6	I	1
8	I	1
10	I	1
12	I	1
14	I	1
	CFD	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมทดสอบที่ 2 เมื่อพิจารณาโครงสร้างเฉพาะมีเพียงหนึ่งโครงสร้างเป็นโครงสร้างควบคุมการไหลของประโยคเงื่อนไข Select Case หรือ กราฟควบคุมการไหลเฉพาะ C ดังนั้นค่า CFS เท่ากับ 1 และจากตารางที่ 4.2 ได้ค่า CFD เท่ากับ 1 เมื่อแทนค่าในสูตร $SDM = CFS + CFD + 1$ จึงได้ค่าความซับซ้อนเท่ากับ $1 + 1 + 1 = 3$

ตารางที่ 4.2 จำนวนข้อมูลควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมทดสอบที่ 2

โหนด	ข้อมูลควบคุมการไหล	CFD _i
2	I	1
	CFD	1

โปรแกรมทดสอบที่ 3 เมื่อพิจารณาโครงสร้างเฉพาะมี 3 โครงสร้าง เป็นโครงสร้างควบคุมการไหลของประโยคเงื่อนไข Select Case หรือ กราฟควบคุมการไหลเฉพาะ C และมีการซ้อนในของโครงสร้างควบคุมการไหลของประโยคเงื่อนไข For . . . Next หรือ กราฟควบคุมการไหลเฉพาะ D₂ 2 โครงสร้าง ดังนั้นค่า CFS เท่ากับ 3 และจากตารางที่ 4.3 ได้ค่า CFD เท่ากับ 3 เมื่อแทนค่าในสูตร $SDM = CFS + CFD + 1$ จึงได้ค่าความซับซ้อนเท่ากับ $3 + 3 + 1 = 7$

ตารางที่ 4.3 จำนวนข้อมูลควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมทดสอบที่ 3

โหนด	ข้อมูลควบคุมการไหล	CFD
2	Index	1
4	i	1
7	i	1
	CFD	3

โปรแกรมทดสอบที่ 4 เมื่อพิจารณาโครงสร้างเฉพาะมี 3 โครงสร้าง เป็นโครงสร้างควบคุมการไหลของประโยคเงื่อนไข Select Case หรือ กราฟควบคุมการไหลเฉพาะ C ต่อด้วยโครงสร้างควบคุมการไหลของประโยคเงื่อนไข For . . . Next หรือ กราฟควบคุมการไหลเฉพาะ D₂ และมีการซ้อนในโครงสร้างควบคุมการไหลของประโยคเงื่อนไข If . . . Else หรือ กราฟควบคุมการไหลเฉพาะ D₁ ดังนั้นค่า CFS เท่ากับ 3 และจากตารางที่ 4.4 ได้ค่า CFD เท่ากับ 4 จึงได้ค่าความซับซ้อนเท่ากับ $3 + 4 + 1 = 8$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 จำนวนข้อมูลควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมทดสอบที่ 4

โหนด	ข้อมูลควบคุมการไหล	CFD
2	tabMtxFile.ColumnText(11)	1
8	i	1
9	cmdMtxFile().Enabled	1
	i	1
CFD		4

โปรแกรมทดสอบที่ 5 เมื่อพิจารณาโครงสร้างเฉพาะมี 2 โครงสร้าง เป็นโครงสร้างควบคุมการไหลของประโยคเงื่อนไข Select Case หรือ กราฟควบคุมการไหลเฉพาะ C ซ้อนในด้วยโครงสร้างควบคุมการไหลของประโยคเงื่อนไข If . . . ElseIf หรือ กราฟควบคุมการไหลเฉพาะ D₄ ดังนั้นค่า CFS เท่ากับ 2 และจากตารางที่ 4.5 ได้ค่า CFD เท่ากับ 4 จึงได้ค่าความซับซ้อนเท่ากับ $2 + 4 + 1 = 7$

ตารางที่ 4.5 จำนวนข้อมูลควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมทดสอบที่ 5

โหนด	ข้อมูลควบคุมการไหล	CFD
2	mAddUp_flag	1
4	mRecordAmt	1
6	mRecordNo	1
8	mRecordNo	1
CFD		4

โปรแกรมทดสอบที่ 6 เมื่อพิจารณาโครงสร้างเฉพาะมี 9 โครงสร้าง เป็นโครงสร้างควบคุมการไหลของประโยคเงื่อนไข For หรือ กราฟควบคุมการไหลเฉพาะ D₂ ทั้งหมด ดังนั้นค่า CFS เท่ากับ 9 และจากตารางที่ 4.6 ได้ค่า CFD เท่ากับ 9 จึงได้ค่าความซับซ้อนเท่ากับ $9 + 9 + 1 = 19$

ตารางที่ 4.6 จำนวนข้อมูลควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมทดสอบที่ 6

โหนด	ข้อมูลควบคุมการไหล	CFD
2	i	1
3	j	1
7	i	1
8	j	1
12	i	1
13	j	1
16	i	1
17	j	1
18	k	1
	CFD	9

4.7 การเปรียบเทียบตัววัด SDM กับตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo

จากโปรแกรมทดสอบที่ 1 และ โปรแกรมทดสอบที่ 2 ซึ่งมีผลลัพธ์การทำงานเหมือนกัน แต่ใช้โครงสร้างการเขียนโปรแกรมไม่เหมือนกัน แสดงผลการวัดความซับซ้อนเปรียบเทียบตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM ได้ดังตารางที่ 4.7

ตารางที่ 4.7 ผลการวัดความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM ของโปรแกรมทดสอบที่ 1 กับโปรแกรมทดสอบที่ 2

ตัววัด	ความซับซ้อนโปรแกรมทดสอบที่ 1	ความซับซ้อนโปรแกรมทดสอบที่ 2
Halstead	$12 \times 24 \times (45 + 24) \log_2(12 + 17) / 2 \times 17$ = 2,839.347	$11 \times 18 \times (33 + 18) \log_2(11 + 17) / 2 \times 17$ = 1,427.784
McCabe	$25 - 19 + 2 = 8$	$19 - 13 + 2 = 8$
Oviedo	$25 + 7 = 32$	$19 + 1 = 20$
SDM	$1 + 7 + 1 = 9$	$1 + 1 + 1 = 3$

จากตารางที่ 4.7 แสดงผลการวัดความซับซ้อนของโปรแกรมทดสอบที่ 1 กับโปรแกรมทดสอบที่ 2 ที่ได้จากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM พบว่าผลที่วัดได้จากทุกตัววัดเป็นไปในทำนองเดียวกันคือ ค่าความซับซ้อนของโปรแกรมทดสอบที่ 1 จะให้ค่าความซับซ้อนมากกว่าโปรแกรมทดสอบที่ 2 ยกเว้นตัววัดของ McCabe ค่าความซับซ้อนที่ได้ของทั้งสองโปรแกรมเท่ากัน ซึ่งขัดแย้งกับความเป็นจริงและตามหลักการเขียนโปรแกรมแล้ว การใช้คำสั่งเงื่อนไขแบบ Select Case มีความซับซ้อนน้อยกว่าการเขียนโปรแกรมโดยใช้เงื่อนไข If... ElseIf

จากตัวอย่างนี้แสดงให้เห็นว่าตัววัด SDM ให้ค่าความซับซ้อนได้ตรงตามความเป็นจริงมากกว่าตัววัดของ McCabe ซึ่งผลการวัดยังคงเป็นไปในทำนองเดียวกับตัววัดของ Halstead และตัววัดของ Oviedo ผู้วิจัยจึงจะได้นำเสนอตัวอย่างต่อไปเพื่อเปรียบเทียบให้เห็นว่าตัววัด SDM ให้ค่าความซับซ้อนได้ตรงตามความเป็นจริงมากกว่าตัววัดอื่น ๆ ซึ่งได้แสดงโปรแกรมที่ใช้เปรียบเทียบไว้ในรูปที่ 4.4

```
Sub cmdLeft_Click ()
```

```
  If IsDate(d) Then
```

```
    d = DateAdd("d", -1, d)
```

```
    lblDate.Caption = "Date : " & Format$(d, "dddd mmmm d, yyyy")
```

```
    txtCallDate.Text = Format$(d, "DD-MM-YYYY")
```

```
    lCallDate = txtCallDate.Text
```

```
    Call cmdMtx_Click(OIndex)
```

```
    If Trim$(tabMtxId.ColumnText(1)) <> "" Then
```

```
      Call cmdRun_Click
```

```
    End If
```

```
  Else
```

```
    lCallDate = ""
```

```
    txtCallDate.Text = " - - - - "
```

```
  End If
```

```
End Sub
```

(ก)

```
Sub cmdMenu_Click(Index As Integer)
```

```
  Select Case Index
```

```
    Case 0
```

```
      frmFilter.Show
```

```
    Case 1
```

```
      frmPricing.Show
```

```
    Case 2
```

```
      frmEditMtxFile.Show
```

```
    Case 3
```

```
      Unload Me
```

```
  End Select
```

```
End Sub
```

(ข)

รูปที่ 4.4 โปรแกรมเปรียบเทียบความซับซ้อนจากตัววัด SDM กับตัววัดอื่น

```

Function ISumTotalCall (cdate As String, mtxid As String)
  Dim Snap As SnapShot
  Dim sql As String
  sql = "SELECT SUM(TOTAL_CALL) FROM MTX_FILE_DETAIL "
  If cdate <> "" Or mtxid <> "" Then
    sql = sql & "WHERE "
  End If
End Function

```

(ก)

```

Function ISumTotalCall (cdate As String, mtxid As String)
  Dim Snap As SnapShot
  Dim sql As String
  sql = "SELECT SUM(TOTAL_CALL) "
  sql = sql & "FROM MTX_FILE_DETAIL "
  If cdate <> "" Or mtxid <> "" Then
    sql = sql & "WHERE "
  End If
End Function

```

(ข)

รูปที่ 4.4 โปรแกรมเปรียบเทียบความซับซ้อนจากตัววัด SDM กับตัววัดอื่น (ต่อ)

จากโปรแกรมรูปที่ 4.4 (ก) และ (ข) แสดงผลการวัดความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM ได้ดังตารางที่ 4.8

ตารางที่ 4.8 ผลการวัดความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM ของโปรแกรมรูปที่ 4.4 (ก) และ (ข)

ตัววัด	ความซับซ้อนโปรแกรมรูปที่ 4.4 (ก)	ความซับซ้อนโปรแกรมรูปที่ 4.4 (ข)
Halstead	$21 \times 22 \times (53 + 22) \log_2(21 + 14) / 2 \times 14$ $= 6,347.488$	$11 \times 10 \times (18 + 10) \log_2(11 + 9) / 2 \times 9$ $= 739.530$
McCabe	$9 - 8 + 2 = 3$	$10 - 8 + 2 = 4$
Oviedo	$10 + 0 = 10$	$10 + 1 = 11$
SDM	$2 + 4 + 1 = 7$	$1 + 1 + 1 = 3$

จากตารางที่ 4.8 แสดงผลการวัดความซับซ้อนของโปรแกรมรูปที่ 4.4 (ก) และ (ข) ที่ได้ จากตัววัดของ Halstead McCabe Oviedo และตัววัด SDM พบว่าค่าความซับซ้อนจากตัววัด ของ McCabe และตัววัดของ Oviedo โปรแกรมรูปที่ 4.4 (ก) ซับซ้อนน้อยกว่าโปรแกรมรูปที่ 4.4 (ข) ส่วนค่าความซับซ้อนจากตัววัดของ Halstead และตัววัด SDM โปรแกรมรูปที่ 4.4 (ก) จะให้ ค่าความซับซ้อนมากกว่าโปรแกรมรูปที่ 4.4 (ข) ซึ่งเป็นตามความรู้สึกจริงของมนุษย์กล่าวคือ โปรแกรมรูปที่ 4.4 (ก) น่าจะมีความซับซ้อนมากกว่าโปรแกรมรูปที่ 4.4 (ข)

จากโปรแกรมรูปที่ 4.4 (ก) และ (ข) พบว่าค่าความซับซ้อนของโปรแกรมจากตัววัดของ McCabe และตัววัดของ Oviedo ให้ผลของความซับซ้อนไม่เป็นไปตามความรู้สึกจริงเนื่องจาก ตัววัดของ McCabe มีข้อจำกัดในการวัด กรณีที่ใช้ Select Case ตามที่กล่าวมาแล้วจากตัวอย่างข้าง ดัน ส่วนตัววัดของ Oviedo ที่ให้ผลคลาดเคลื่อน เนื่องจากประสิทธิภาพของตัววัดนี้จะลดลงในกรณี ที่มีการใช้ตัวแปรแต่ตัวแปรนั้นไม่ได้มีการกำหนดค่าไว้ก่อนหน้า ดังในโปรแกรมที่ 4.4 (ก) มีการ ใช้ตัวแปร d แต่ไม่มีการกำหนดค่าตัวแปร d ไว้ก่อนหน้า โหนดที่มีการใช้ ทำให้ค่า df เป็น 0

จากโปรแกรมรูปที่ 4.4 (ค) และ (ง) แสดงผลการวัดความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM ได้ดังตารางที่ 4.9

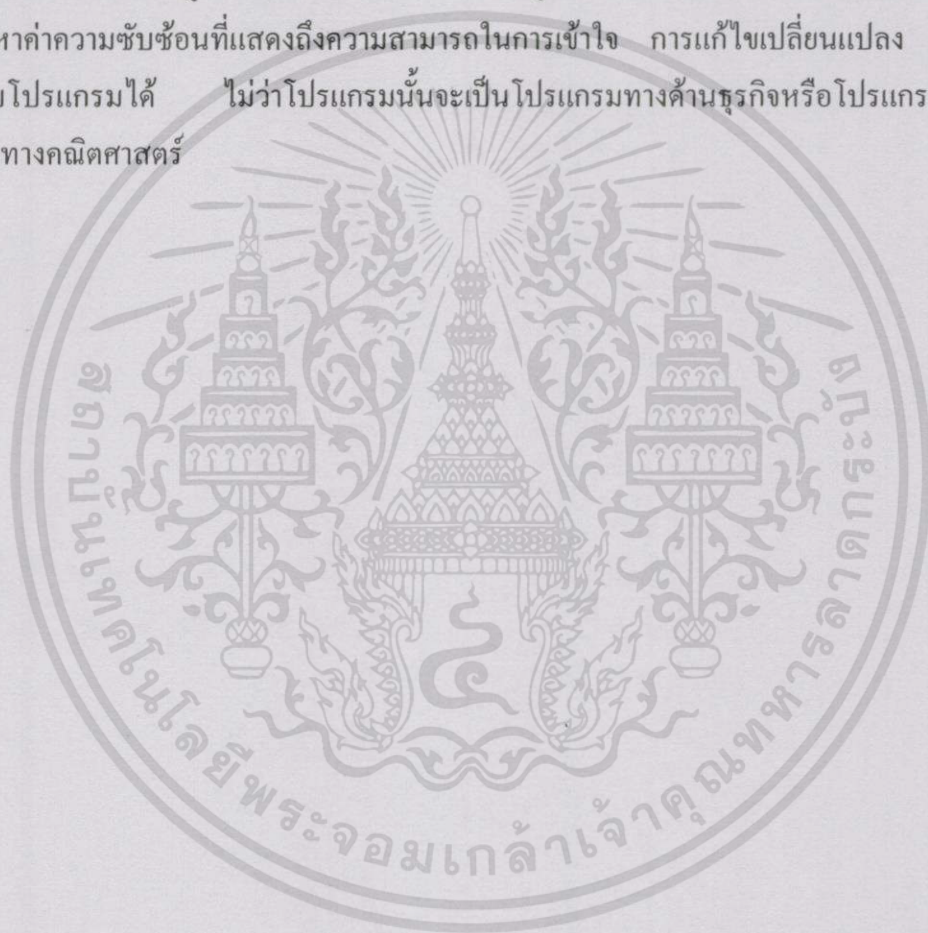
ตารางที่ 4.9 ผลการวัดความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM ของโปรแกรมรูปที่ 4.4 (ค) และ (ง)

ตัววัด	ความซับซ้อนโปรแกรมรูปที่ 4.4 (ค)	ความซับซ้อนโปรแกรมรูปที่ 4.4 (ง)
Halstead	$14 \times 11 \times (21 + 11) \log_2(14 + 6) / 2 \times 6$ = 1,774.872	$14 \times 14 \times (24 + 14) \log_2(14 + 7) / 2 \times 7$ = 2,336.713
McCabe	$6 - 6 + 2 = 2$	$6 - 6 + 2 = 2$
Oviedo	$6 + 3 = 9$	$6 + 3 = 9$
SDM	$1 + 3 + 1 = 4$	$1 + 3 + 1 = 4$

โปรแกรมรูปที่ 4.4 (ค) และ (ง) ให้ผลลัพธ์การทำงานเหมือนกัน แตกต่างที่คำสั่งในการ กำหนดค่าให้ตัวแปร sql ซึ่งตารางที่ 4.9 แสดงผลการวัดความซับซ้อนของโปรแกรมรูปที่ 4.4 (ค) และ (ง) ที่ได้จากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM พบว่าค่าความซับซ้อนจากตัววัดของ Halstead แตกต่างจากตัววัดอื่น ๆ ค่าความซับซ้อนตัววัดของ Halstead จากโปรแกรมรูปที่ 4.4 (ค) ซับซ้อนกว่าโปรแกรมรูปที่ 4.4 (ง) ส่วนตัววัดอื่นค่าความ ซับซ้อนของทั้งสองโปรแกรมเท่ากัน

จากการเปรียบเทียบผลการวัดความซับซ้อนจากตัววัดทั้ง 4 วิธี พบว่าตัววัด SDM จะให้ค่าความซับซ้อนที่ตรงตามความเป็นจริงในแง่ความรู้สึกรู้สึกของมนุษย์ และแก้ไขจุดอ่อนจากข้อจำกัดของตัววัดทั้ง 3 วิธีได้ด้วย

จากโปรแกรมทดสอบ 5 โปรแกรมแรกเป็นโปรแกรมระบบงานทางธุรกิจ โปรแกรมทดสอบที่ 6 เป็นโปรแกรมที่มีการคำนวณทางคณิตศาสตร์ ซึ่งจากผลการวัดความซับซ้อนจากตัววัดทั้ง 4 วิธี ให้ค่าความซับซ้อนของโปรแกรมที่มีการคำนวณทางคณิตศาสตร์สูงกว่าโปรแกรมระบบงานธุรกิจแสดงไว้ในตัวอย่างข้างต้น ซึ่งแนวโน้มความซับซ้อนของโปรแกรมที่มีการคำนวณทางคณิตศาสตร์น่าจะสูงกว่าโปรแกรมระบบงานทางธุรกิจ ดังนั้นตัววัดทั้ง 4 วิธีสามารถนำไปใช้ในการหาค่าความซับซ้อนที่แสดงถึงความสามารถในการเข้าใจ การแก้ไขเปลี่ยนแปลง และการทดสอบโปรแกรมได้ ไม่ว่าจะโปรแกรมนั้นจะเป็นโปรแกรมทางด้านธุรกิจหรือโปรแกรมที่มีการคำนวณทางคณิตศาสตร์



บทที่ 5

ผลการทดสอบการวัดความซับซ้อนของซอฟต์แวร์

5.1 ข้อมูลที่ใช้ในการทดสอบ

ข้อมูลที่นำมาทดสอบเป็นระบบงานที่พัฒนาขึ้นด้วยภาษาวิซวลเบสิกซึ่งใช้งานจริงมีทั้งหมด 7 ระบบ แสดงจำนวนไฟล์และโพรซีเจอร์ทั้งหมดของแต่ละระบบได้ดังตารางที่ 5.1

ตารางที่ 5.1 จำนวนไฟล์และโพรซีเจอร์ของโปรแกรมทดสอบ

ระบบ	ชื่อระบบงาน	จำนวนไฟล์	จำนวนโพรซีเจอร์
1.	ระบบตรวจสอบยอดค้างชำระของลูกค้า	7	115
2.	ระบบบันทึกปัญหาและการแก้ปัญหาไปยังผู้บริหาร	5	34
3.	ระบบประเมินข้อมูลแหล่งท่องเที่ยวเบื้องต้น	61	526
4.	ระบบข้อมูลฝ่ายบุคคล	17	127
5.	ระบบจัดเก็บเอกสาร	20	184
6.	ระบบจัดเก็บข้อมูล	1	39
7.	ระบบสาริตและการจัดเก็บและเรียกค้นการ์ดลายเซ็น	6	39
รวม		117	1064

5.2 ผลการทดสอบการวัดค่าความซับซ้อน

ผลการทดสอบการวัดค่าความซับซ้อนของซอฟต์แวร์จะแสดงค่าความซับซ้อนของแต่ละโพรซีเจอร์โดยตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM จากงานวิจัยนี้ แสดงไว้ในภาคผนวก ก. โดยเรียงลำดับตามระบบงานจากตารางที่ 5.1

จากผลการทดลองพบว่าค่าความซับซ้อนที่ได้จากตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM เป็นระบบจำนวนนับ ส่วนตัววัดของ Halstead เป็นระบบจำนวนจริง และผลที่ได้ส่วนใหญ่จะเป็นในทำนองเดียวกันคือ โปรแกรมที่มีความซับซ้อนมากกว่าจะให้ค่าความซับซ้อนมากกว่าโปรแกรมที่มีความซับซ้อนน้อยกว่า จะมีบางกรณีที่ผลลัพธ์ของตัววัดบางตัวขัดแย้งกับตัววัดอื่น เกิดจากข้อจำกัดของตัววัดนั้น เมื่อพิจารณาตัววัด SDM ผลลัพธ์เป็นในทำนองเดียวกับตัววัดอื่น ซึ่งตรงตามความเป็นจริง ค่าความซับซ้อนนี้เป็นเพียงตัวเลข ไม่สามารถบอกได้ว่าโปรแกรมที่วัดมีประสิทธิภาพอย่างไร นอกจากจะต้องเปรียบเทียบค่ากับโปรแกรมอื่น ยกเว้นค่าความซับซ้อนที่ได้จากตัววัดของ McCabe สามารถบอกได้ว่าโปรแกรมใดมีความเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

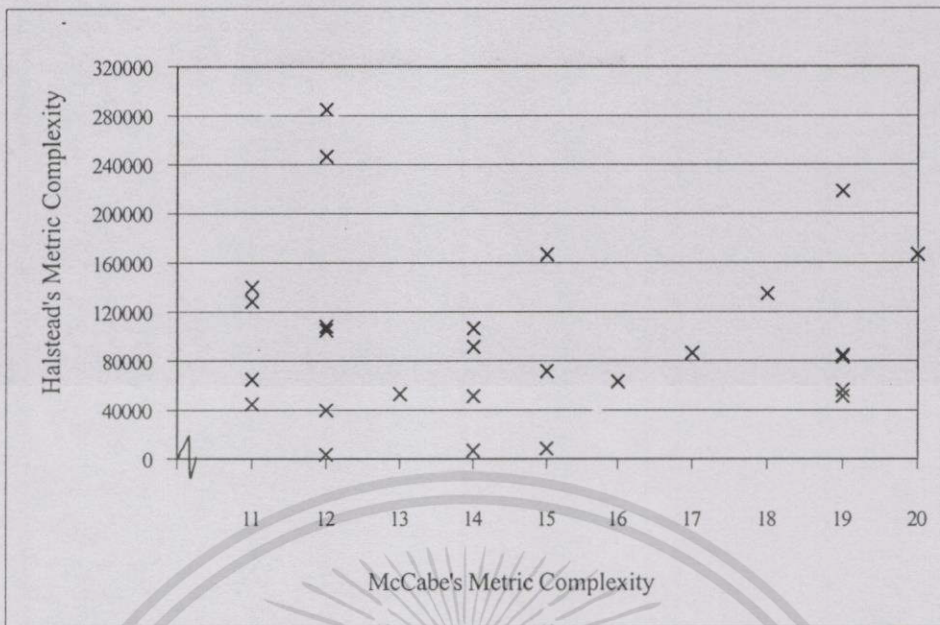
เหมาะสม และโปรแกรมใดควรที่จะได้รับการแก้ไข เนื่องจาก McCabe ได้กำหนดเกณฑ์ความซับซ้อนที่เหมาะสมว่าแต่ละโพรซีเจอร์ควรมีค่า $v(F)$ ไม่เกิน 10

ดังนั้นจากผลการทดลองนี้ผู้วิจัยจึงได้นำเสนอเกณฑ์ความซับซ้อนที่เหมาะสมสำหรับตัววัดของ Halstead ตัววัดของ Oviedo และตัววัด SDM โดยเปรียบเทียบกับเกณฑ์ความซับซ้อนจากตัววัดของ McCabe ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อต่อไป

5.3 เกณฑ์ความซับซ้อนที่เหมาะสมของโปรแกรม

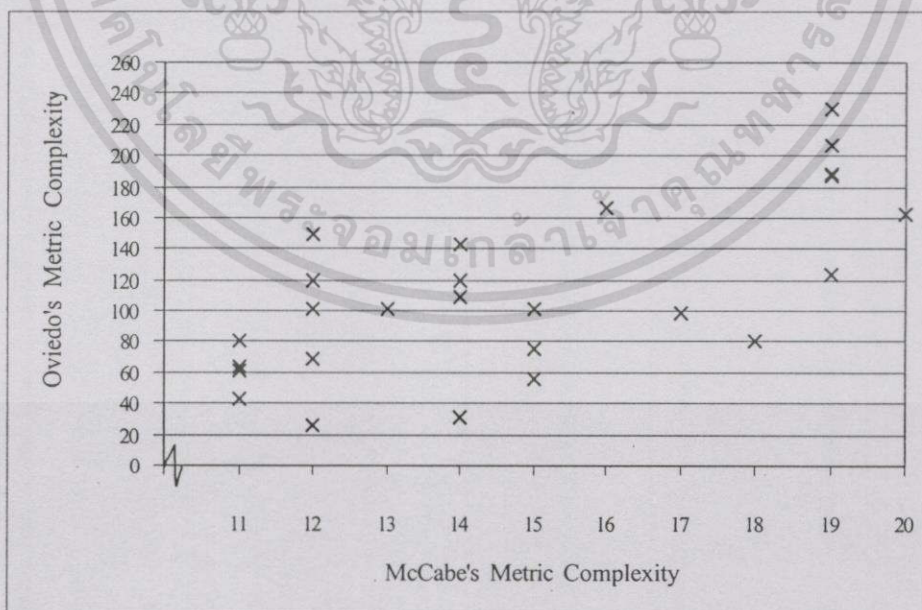
ในการวัดค่าความซับซ้อนที่ได้จากตัววัดต่าง ๆ พบว่ามีเพียงตัววัดของ McCabe เท่านั้นที่มีการกำหนดเกณฑ์การวัดที่เหมาะสมคือ McCabe แนะนำว่าค่าความซับซ้อนที่เหมาะสมควรอยู่ระหว่าง 3 ถึง 7 ส่วนตัววัดอื่น ๆ ไม่มีการกำหนดเกณฑ์ที่เหมาะสม ดังนั้นเมื่อวัดค่าความซับซ้อนได้ จะบอกได้เพียงแค่ว่าโปรแกรมนั้นมีค่าความซับซ้อนน้อยหรือซับซ้อนมาก หรือจะต้องนำไปเปรียบเทียบกับโพรซีเจอร์อื่น ๆ ไม่สามารถบอกได้ว่าโปรแกรมนั้นมีความซับซ้อนอยู่ในระดับที่เหมาะสมหรือไม่ ดังนั้นผู้วิจัยจึงได้นำเสนอเกณฑ์ความซับซ้อนที่เหมาะสมโดยเปรียบเทียบกับเกณฑ์ของ McCabe จากที่ McCabe กล่าวว่าเกณฑ์ความซับซ้อนที่เหมาะสมควรอยู่ระหว่าง 3 ถึง 7 และไม่ควรมีความซับซ้อนเกิน 10 ดังนั้นโปรแกรมที่มีความซับซ้อนของ McCabe เกิน 10 จึงน่าที่จะมีความซับซ้อนมากเกินไป ซึ่งผู้วิจัยได้นำผลการวัดความซับซ้อนที่ได้จากตารางที่ 5.2 – 5.8 แสดงกราฟเปรียบเทียบค่าความซับซ้อนตัววัดของ Halstead ตัววัดของ Oviedo และตัววัด SDM กับตัววัดของ McCabe เพื่อกำหนดเกณฑ์ความซับซ้อนที่เหมาะสม ดังแสดงในรูปที่ 5.1 รูปที่ 5.2 และรูปที่ 5.3 ตามลำดับ

จากกราฟรูปที่ 5.1 แสดงค่าความซับซ้อนจากตัววัดของ Halstead เปรียบเทียบกับตัววัดของ McCabe ซึ่งพิจารณาความซับซ้อนจากตัววัดของ McCabe ที่มากกว่า 10 พบว่าค่าความซับซ้อนจากตัววัดของ Halstead มีค่ามากกว่า 40,000 อยู่ร้อยละ 93.48 ดังนั้นเกณฑ์ความซับซ้อนที่เหมาะสมสำหรับตัววัดของ Halstead ไม่ควรเกิน 40,000



รูปที่ 5.1 กราฟแสดงค่าความซับซ้อนจากตัววัดของ Halstead เปรียบเทียบกับตัววัดของ McCabe

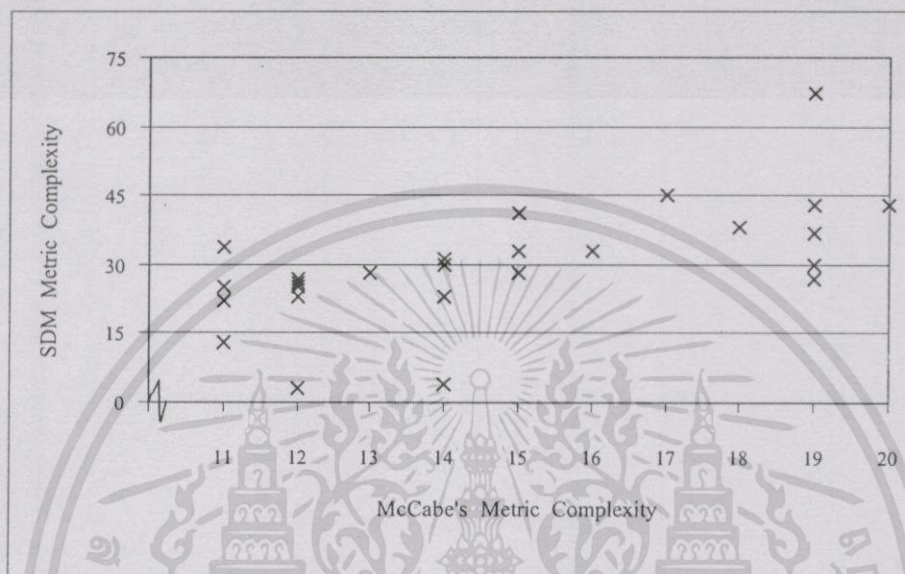
จากกราฟรูปที่ 5.2 แสดงค่าความซับซ้อนจากตัววัดของ Oviedo เปรียบเทียบกับตัววัดของ McCabe ซึ่งพิจารณาค่าความซับซ้อนจากตัววัดของ McCabe ที่มากกว่า 10 พบว่าค่าความซับซ้อนจากตัววัดของ Oviedo มีค่ามากกว่า 40 อยู่ร้อยละ 95.65 ดังนั้นเกณฑ์ความซับซ้อนที่เหมาะสมสำหรับตัววัดของ Oviedo ไม่ควรเกิน 40



รูปที่ 5.2 กราฟแสดงค่าความซับซ้อนจากตัววัดของ Oviedo เปรียบเทียบกับตัววัดของ McCabe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกราฟรูปที่ 5.3 แสดงค่าความซับซ้อนจากตัววัด SDM เปรียบเทียบกับตัววัดของ McCabe ซึ่งพิจารณาค่าความซับซ้อนจากตัววัดของ McCabe ที่มากกว่า 10 พบว่าค่าความซับซ้อนจากตัววัด SDM มีค่ามากกว่า 15 อยู่ร้อยละ 93.48 ดังนั้นเกณฑ์ความซับซ้อนที่เหมาะสมสำหรับตัววัด SDM ไม่ควรเกิน 15

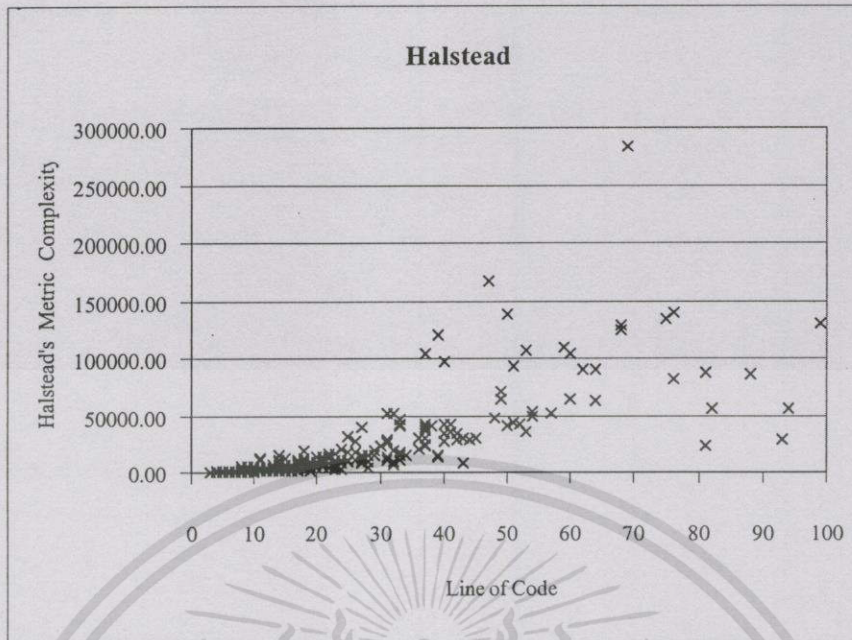


รูปที่ 5.3 กราฟแสดงค่าความซับซ้อนจากตัววัดของตัววัด SDM เปรียบเทียบกับตัววัดของ McCabe

ถ้าโปรแกรมใดมีค่าความซับซ้อนมากกว่าเกณฑ์ที่กำหนดหมายความว่าโปรแกรมนั้นมีความซับซ้อนมากเกินไป ควรได้รับการแก้ไขหรือเปลี่ยนแปลง ซึ่งในบางครั้งการแก้ไขหรือเปลี่ยนแปลงอาจพบว่าโปรแกรมนั้นมีขนาดลดลงแต่ทำให้ความซับซ้อนของโครงสร้างเพิ่มขึ้น คือค่าความซับซ้อนจากตัววัดของ Halstead ลดลงแต่อาจทำให้ค่าความซับซ้อนจากตัววัดของ McCabe เพิ่มขึ้น ดังนั้นตัววัด SDM น่าจะเหมาะสมสำหรับกำหนดว่าโปรแกรมนั้นควรได้รับการแก้ไขหรือเปลี่ยนแปลงหรือไม่ เนื่องจากตัววัดนี้จะพิจารณาทั้งจากโครงสร้างและข้อมูลประกอบกัน

5.4 ความสัมพันธ์ของความซับซ้อนกับจำนวนบรรทัดของโปรแกรม

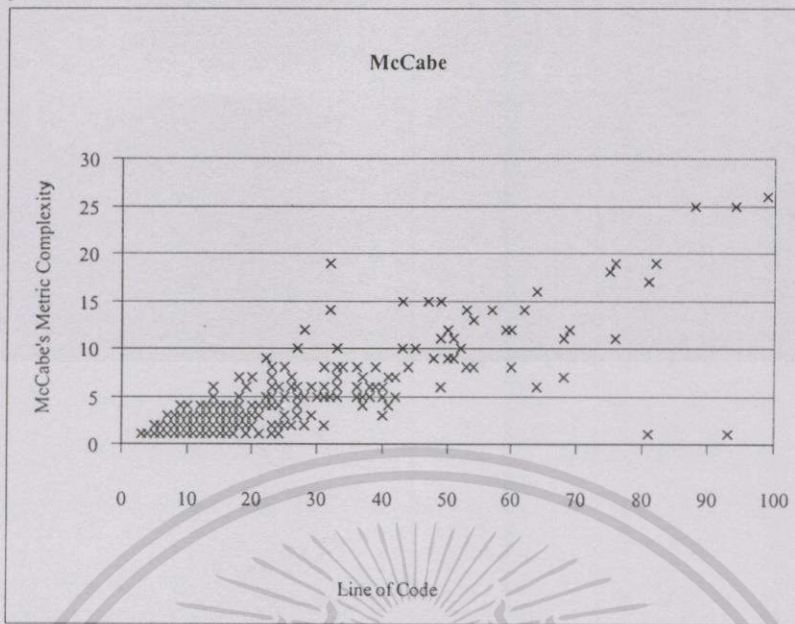
ในส่วนนี้เราจะแสดงให้เห็นว่าค่าความซับซ้อนของตัววัดแต่ละวิธี มีความสัมพันธ์กับจำนวนบรรทัดของโปรแกรมหรือไม่ โดยพิจารณาค่าความซับซ้อนที่ได้จากโปรแกรมระบบงานทางธุรกิจดังแสดงไว้ในภาคผนวก ก. ซึ่งแสดงความสัมพันธ์ของความซับซ้อนจากตัววัดแต่ละวิธีกับจำนวนบรรทัดของโปรแกรมหดกราฟรูปที่ 5.4 – 5.7



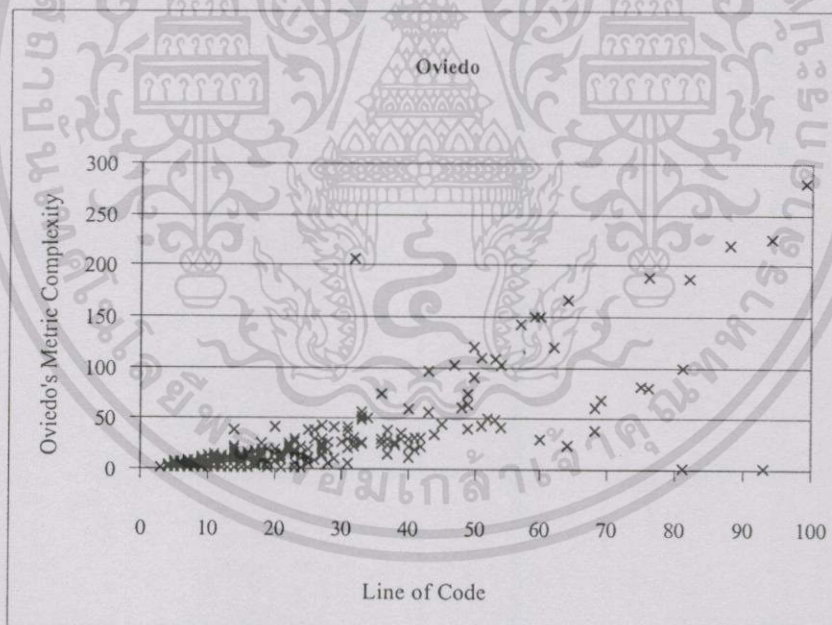
รูปที่ 5.4 กราฟแสดงความสัมพันธ์ของความซับซ้อนจากตัววัดของ Halstead กับจำนวนบรรทัดของโปรแกรม

จากกราฟรูปที่ 5.4 แสดงให้เห็นว่าความสัมพันธ์ของความซับซ้อนจากตัววัดของ Halstead กับจำนวนบรรทัดของโปรแกรมมีความสัมพันธ์ในแนวทางเดียวกัน กล่าวคือ โปรแกรมที่มีจำนวนบรรทัดมากจะให้ค่าความซับซ้อนของโปรแกรมจากตัววัดของ Halstead มากด้วย เนื่องจาก Halstead พิจารณาความซับซ้อนจากขนาดของโปรแกรม ดังนั้นจำนวนมากจึงส่งผลให้ความซับซ้อนจากตัววัดนี้มากด้วย

จากกราฟรูปที่ 5.5 - 5.7 แสดงให้เห็นความสัมพันธ์ของความซับซ้อนจากตัววัดของ McCabe ตัววัดของ Oviedo และตัววัดSDM ซึ่งเป็นกลุ่มตัววัดที่มีการพิจารณาจากโครงสร้างของโปรแกรม ดังนั้นความซับซ้อนจึงไม่ขึ้นอยู่กับจำนวนบรรทัดเท่าใดนัก แต่ก็มีแนวโน้มที่จำนวนบรรทัดส่งผลต่อค่าความซับซ้อนอยู่บ้าง กล่าวคือ ถ้าจำนวนบรรทัดน้อยความซับซ้อนของโปรแกรมก็จะน้อยด้วย แต่ถ้าจำนวนบรรทัดมากค่าความซับซ้อนจากตัววัดทั้ง 3 ไม่ได้มากตามไปด้วยเสมอ

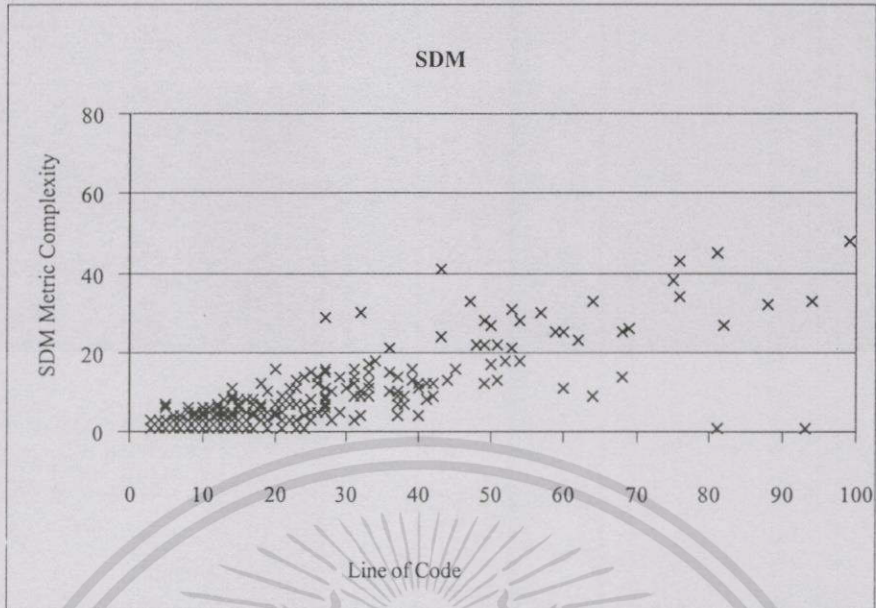


รูปที่ 5.5 กราฟแสดงความสัมพันธ์ของความซับซ้อนจากตัววัดของ McCabe กับจำนวนบรรทัดของโปรแกรม



รูปที่ 5.6 กราฟแสดงความสัมพันธ์ของความซับซ้อนจากตัววัดของ Oviedo กับจำนวนบรรทัดของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 กราฟแสดงความสัมพันธ์ของความซับซ้อนจากตัววัด SDM กับจำนวนบรรทัดของโปรแกรม

บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 บทสรุป

ความซับซ้อนของซอฟต์แวร์ (Software Complexity) หมายถึง ความยากต่อการเข้าใจ การแก้ไขเปลี่ยนแปลง และการทดสอบโปรแกรม [3] เนื่องจากความซับซ้อนส่งผลต่อความน่าเชื่อถือ การบำรุงรักษา การทดสอบ และการแก้ไขเปลี่ยนแปลงก็จะยุ่งยากขึ้น และคุณสมบัติเหล่านี้เป็นปัจจัยที่ส่งผลต่อต้นทุนและเวลาในการพัฒนาซอฟต์แวร์ด้วย

ในปี 1972 Halstead เป็นผู้เริ่มกำหนดวิธีการมาตรฐานเพื่อวัดความซับซ้อนของโปรแกรม เรียกว่า ตัววัดซอฟต์แวร์ ซึ่งเป็นสาขาย่อยของวิทยาการคอมพิวเตอร์ ต่อมามีนักวิจัยหลายท่าน กำหนดตัววัดขึ้นมามากมาย โดยกำหนดเกณฑ์การพิจารณาที่แตกต่างกัน

งานวิจัยเกี่ยวกับตัววัดความซับซ้อนของซอฟต์แวร์ที่ได้วิจัยกันมาแบ่งออกเป็น 2 ประเภทใหญ่ ๆ คือ แบบสแตติก และตัววัดแบบไดนามิก ตัววัดแบบสแตติกเป็นการวัดความซับซ้อนโดยวิเคราะห์จากรหัสต้นฉบับ (Source Code) ซึ่งปัจจัยที่ใช้เป็นคุณสมบัติที่สามารถวัดได้โดยตรงจากรหัสต้นฉบับ เช่น จำนวนบรรทัดของโปรแกรม จำนวนตัวแปร จำนวนโครงสร้างตรรกะ เป็นต้น ส่วนตัววัดแบบไดนามิกเป็นการวัดความซับซ้อนในช่วงเวลาที่โปรแกรมทำงาน ซึ่งปัจจัยที่ใช้เป็นคุณสมบัติที่วัดได้โดยพิจารณาขณะที่โปรแกรมทำงาน เช่น การวัดค่า Big-O ในงานวิจัยนี้ศึกษาตัววัดเฉพาะแบบสแตติก ซึ่งจะพิจารณาในส่วนที่เป็น Procedural

งานวิจัยที่เกี่ยวกับตัววัดแบบสแตติกสามารถแบ่งได้เป็น 4 ลักษณะ คือ

1. **ตัววัดขนาด (Size Metrics)** เป็นวิธีวัดที่ง่าย และนิยมใช้กันมาก ใช้ขนาดเป็นเกณฑ์บอกถึงความซับซ้อนของโปรแกรม กล่าวคือ โปรแกรมที่มีขนาดใหญ่กว่าจะมีความซับซ้อนมากกว่าโปรแกรมที่มีขนาดเล็กกว่า ตัววัดที่จัดอยู่ในประเภทนี้ ได้แก่ ตัววัดจำนวนบรรทัดของโปรแกรม (Line of code) และตัววัดของ Halstead (Halstead's Metric) พิจารณาจากโอเปอเรเตอร์และโอเปอเรนด์ นำไปใช้ในการวัดความยาวของโปรแกรม การวัดปริมาตร และการวัดระดับของโปรแกรม ต่อมา Halstead ได้กำหนดสูตรเพื่อใช้ในการวัดความพยายามของโปรแกรม ซึ่งสามารถแสดงถึงความซับซ้อนได้ กล่าวคือโปรแกรมใดที่มีค่าความพยายามมาก โปรแกรมนั้นจะมีความซับซ้อนมากด้วย

2. **ตัววัดโครงสร้างตรรกะ (Logic Structure Metrics)** หรือตัววัดโครงสร้างการควบคุมการไหล (Control Flow Structure Metrics) เป็นการหาค่าความซับซ้อนจากกราฟของโปรแกรม เรียกกราฟนี้ว่า กราฟควบคุมการไหล (Control Flow Graphs) ตัววัดที่จัดอยู่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทนี้ ได้แก่ ตัววัดค่าไซโคลเมตริกของ McCabe (McCabe's Cyclomatic Metric) ตัววัดปม (Knot Metric) และตัวขอบเขต (Scope Metric) ตัววัดของ Chen (Chen's Metric) ตัววัดของ McClure (McClure's Metric) เป็นต้น ตัววัดในกลุ่มนี้ส่วนใหญ่มีแนวความคิดเพื่อที่จะแก้ไขข้อจำกัดตัววัดของ McCabe ซึ่งโปรแกรมที่มีความซับซ้อนมากจะให้ค่าความซับซ้อนมากกว่าโปรแกรมที่มีความซับซ้อนน้อยกว่า

3. **ตัววัดข้อมูล (Data Metrics)** วิธีนี้คำนวณค่าความซับซ้อนโดยพิจารณาจาก การสื่อสารและการใช้ข้อมูลร่วมกันระหว่างโมดูล การจัดการและการกระจายของข้อมูล ค่าคงที่ และตัวแปรต่าง ๆ ภายในโปรแกรม ตัววัดที่จัดอยู่ในประเภทนี้ ได้แก่ ตัววัดของ Henry และ Kafura (Henry and Kafura's Metric) พิจารณาความซับซ้อนจากการสื่อสาร การใช้ข้อมูลร่วมกันระหว่างโมดูล ความยาวของโปรแกรม ส่วนตัววัดของ Shepperd และ Ince (Shepperd and Ince's Metrics) ประยุกต์จากตัววัดของ Henry และ Kafura ซึ่งจะไม่พิจารณาความยาวของโปรแกรม เพราะเขาคิดว่าความซับซ้อนของโปรแกรมไม่ได้ขึ้นอยู่กับความยาวของโปรแกรม

4. **ตัววัดแบบผสม (Hybrid Metrics)** เป็นตัววัดค่าความซับซ้อนจากการไหลของข้อมูล (Data Flow) และจากกราฟควบคุมการไหล ตัววัดที่จัดอยู่ในประเภทนี้ ได้แก่ ตัววัดของ Oviedo (Oviedo's Metric)

จากการศึกษาตัววัดทั้ง 4 ลักษณะพบว่าตัววัดโครงสร้างตรรกะได้รับความสนใจจากนักวิจัยจำนวนมาก เนื่องจากนักวิจัยเชื่อว่าความซับซ้อนของโปรแกรมน่าจะเกิดจากโครงสร้างของโปรแกรมที่ซับซ้อนมากกว่าเกิดจากโปรแกรมที่มีขนาดใหญ่เท่านั้น ต่อมา Oviedo เห็นว่าความซับซ้อนของโปรแกรมไม่น่าเกิดจากโครงสร้างเพียงอย่างเดียวเท่านั้น เขาจึงพิจารณาความซับซ้อนจากโครงสร้างและการไหลของข้อมูล

ในงานวิจัยนี้ผู้วิจัยนำตัววัดความซับซ้อน 3 แบบมาทำการทดลองวัดค่าความซับซ้อน ตัววัดแรกคือตัววัดของ Halstead จะพิจารณาจากโอเปอเรเตอร์และโอเปอเรนด์ ซึ่งเป็นตัววัดขนาดตัววัดที่ 2 คือ ตัววัดของ McCabe ซึ่งเป็นการวัดโครงสร้างควบคุมการไหลหรือเรียกว่ากราฟควบคุมการไหล ส่วนตัววัดที่ 3 เป็นตัววัดของ Oviedo เป็นตัววัดแบบผสมโดยพิจารณาปัจจัยความซับซ้อนจากโครงสร้างควบคุมการไหลและการไหลของข้อมูล

นักวิจัยแต่ละท่านกำหนดเกณฑ์การวัดความซับซ้อนที่แตกต่างกันไม่ว่าจะเป็นการวัดจากโครงสร้างหรือการวัดจากขนาดของโปรแกรม เกิดจากการสังเกตเห็นว่าปัจจัยนั้นมีผลต่อความซับซ้อนของโปรแกรมแล้วนำไปปัจจัยเหล่านั้นมาหาความสัมพันธ์เพื่อนำเสนอแนวทางและวิธีการใหม่ซึ่งเชื่อว่าจะสามารถหาค่าความซับซ้อนของซอฟต์แวร์ได้ดีขึ้น ดังนั้นค่าที่ได้จากแต่ละวิธีที่วัดจากโปรแกรมเดียวกันหรือแม้แต่โปรแกรมต่างโครงสร้างแต่ทำงานเหมือนกัน ได้ค่าที่แตกต่างกัน ซึ่งลักษณะความไม่คงเส้นคงวา (Consistent) แบบนี้ ก่อให้เกิดจุดอ่อนของกระบวนการวัดความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซับซ้อนโดยรวม จึงน่าที่จะมีวิธีการวัดที่เป็นกลางโดยการพิจารณาองค์ประกอบหลาย ๆ ส่วนของโปรแกรม

ผู้วิจัยได้นำเสนอตัววัดความซับซ้อนใหม่ เรียกว่า ตัววัด SDM (Control Flow Structure and Control Flow Data Metric) ซึ่งพิจารณาจากโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล แนวคิดเกิดจากข้อจำกัดในการวัดความซับซ้อนจากโครงสร้างควบคุมการไหลเพียงอย่างเดียวของ McCabe ตัววัด SDM ให้ผลตรงตามความรู้สึกที่เป็นจริงมากกว่าเมื่อเทียบกับตัววัดของ Halstead ตัววัดของ McCabe และตัววัดของ Oviedo ซึ่ง McCabe จะพิจารณาความซับซ้อนเฉพาะโครงสร้างควบคุมการไหลเพียงอย่างเดียวและตัววัดใหม่ยังแก้ไขจุดอ่อนตัววัดของ McCabe ในกรณีที่วัดโครงสร้างของคำสั่ง Select Case วิธีการของ McCabe ถ้า Case เพิ่มขึ้นค่าความซับซ้อนของโครงสร้างจะเพิ่มขึ้นด้วย แต่วิธีการใหม่การเพิ่ม Case จะไม่ส่งผลต่อความซับซ้อนของโครงสร้าง แต่ตัวแปรที่ใช้ควบคุมการไหลจะเป็นตัวที่ส่งผลกระทบต่อความซับซ้อนเพิ่มขึ้น ส่วนตัววัดของ Halstead จะพิจารณาความซับซ้อนจากโอเปอเรเตอร์และโอเปอเรนด์ซึ่งพิจารณาจากขนาดของโปรแกรม

จากตัววัดแบบผสมของ Oviedo จะพิจารณาความซับซ้อนจากโครงสร้างควบคุมการไหลและโครงสร้างการไหลของข้อมูล ในทำนองเดียวกันกับตัววัด SDM แต่ตัววัดของ Oviedo ประสิทธิภาพการวัดจะลดลงถ้าตัวแปรที่ใช้ไม่มีการกำหนดค่าไว้ก่อนในโพสิเจอร์นั้น เนื่องจากตัวแปรที่ใช้เป็นแบบโกลบอล แต่ตัววัด SDM ไม่ว่าตัวแปรที่ใช้จะเป็นแบบโกลบอลหรือโลคอลจะไม่ส่งผลให้ประสิทธิภาพการวัดลดลง

ดังนั้นตัววัด SDM มีประสิทธิภาพในการวัดความซับซ้อนเพิ่มมากขึ้น เนื่องจากค่าความซับซ้อนที่ได้ตรงตามความรู้สึกที่เป็นจริงมากกว่า ซึ่งค่าความซับซ้อนที่ได้จะพิจารณาทั้งจากโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล จึงน่าที่จะดีกว่าการพิจารณาจากโครงสร้างควบคุมการไหลเพียงอย่างเดียวหรือพิจารณาเฉพาะโอเปอเรเตอร์และโอเปอเรนด์เท่านั้น และไม่ทราบว่าโปรแกรมนั้นจะเป็นโปรแกรมทางด้านธุรกิจหรือโปรแกรมที่มีการคำนวณทางคณิตศาสตร์ ก็ให้ค่าความซับซ้อนสอดคล้องกับความเป็นจริง

จากการทดลองวัดความซับซ้อนของระบบงานที่พัฒนาขึ้นด้วยภาษาวิซวลเบสิก 7 ระบบงานมีทั้งหมด 1,064 โพสิเจอร์ ค่าความซับซ้อนที่ได้จากตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM เป็นระบบจำนวนนับ ส่วนตัววัดของ Halstead เป็นระบบจำนวนจริง ซึ่งค่าความซับซ้อนนี้เป็นเพียงตัวเลข ไม่สามารถบอกได้ว่าโปรแกรมนั้นมีประสิทธิภาพอย่างไร นอกจากจะต้องเปรียบเทียบค่ากับโปรแกรมอื่น ยกเว้นค่าความซับซ้อนที่ได้จากตัววัดของ McCabe สามารถบอกได้ว่าโปรแกรมใดมีความเหมาะสม และโปรแกรมใดควรที่จะได้รับการแก้ไข เนื่องจาก McCabe ได้กำหนดเกณฑ์ความซับซ้อนที่เหมาะสมว่าแต่ละโพสิเจอร์ควรมีค่า $v(F)$ ไม่เกิน 10 ดังนั้นจากผลการทดลองนี้ผู้วิจัยจึงได้นำเสนอเกณฑ์ความซับซ้อนที่เหมาะสมสำหรับตัววัดของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Halstead ตัววัดของ Oviedo และตัววัด SDM โดยเปรียบเทียบกับเกณฑ์ความซับซ้อนจากตัววัดของ McCabe ซึ่งสามารถกำหนดเกณฑ์ความซับซ้อนที่เหมาะสมได้ดังนี้ เกณฑ์ความซับซ้อนที่เหมาะสมสำหรับตัววัดของ Halstead ไม่ควรเกิน 40,000 เกณฑ์ความซับซ้อนที่เหมาะสมสำหรับตัววัดของ Oviedo ไม่ควรเกิน 40 และเกณฑ์ความซับซ้อนที่เหมาะสมสำหรับตัววัด SDM ไม่ควรเกิน 15 ถ้าโปรแกรมใดมีค่าความซับซ้อนมากกว่าเกณฑ์ที่กำหนดหมายความว่าโปรแกรมนั้นมีความซับซ้อนมากเกินไป ควรได้รับการแก้ไขหรือเปลี่ยนแปลง ซึ่งในบางครั้งการแก้ไขหรือเปลี่ยนแปลงอาจพบว่าโปรแกรมนั้นมีขนาดลดลงแต่ทำให้ความซับซ้อนของโครงสร้างเพิ่มขึ้น คือค่าความซับซ้อนจากตัววัดของ Halstead ลดลงแต่อาจทำให้ค่าความซับซ้อนจากตัววัดของ McCabe เพิ่มขึ้น ดังนั้นตัววัด SDM น่าจะเหมาะสมสำหรับกำหนดว่าโปรแกรมนั้นควรได้รับการแก้ไขหรือเปลี่ยนแปลงหรือไม่ เนื่องจากตัววัดนี้จะพิจารณาทั้งจากโครงสร้างและข้อมูลประกอบกัน

เมื่อเปรียบเทียบความสัมพันธ์ของตัววัดต่าง ๆ กับจำนวนบรรทัดของโปรแกรม จะพบว่าตัววัดของ Halstead ค่าความซับซ้อนมีความสัมพันธ์กับจำนวนบรรทัดของโปรแกรม กล่าวคือโปรแกรมที่มีจำนวนบรรทัดมากจะให้ค่าความซับซ้อนของโปรแกรมจากตัววัดของ Halstead มากด้วย เนื่องจาก Halstead พิจารณาความซับซ้อนจากขนาดของโปรแกรม ดังนั้นจำนวนมากจึงส่งผลให้ความซับซ้อนจากตัววัดนี้มากด้วย ส่วนตัววัดของ McCabe ตัววัดของ Oviedo และ ตัววัด SDM ซึ่งเป็นกลุ่มตัววัดที่มีการพิจารณาจากโครงสร้างของโปรแกรมหากความซับซ้อนจึงไม่ขึ้นอยู่กับจำนวนบรรทัดเท่าใดนัก แต่ก็มีแนวโน้มที่จำนวนบรรทัดส่งผลต่อค่าความซับซ้อนอยู่บ้าง กล่าวคือ ถ้าจำนวนบรรทัดน้อยความซับซ้อนของโปรแกรมก็จะน้อยด้วย แต่ถ้าจำนวนบรรทัดมากค่าความซับซ้อนจากตัววัดทั้ง 3 ไม่ได้มากตามไปด้วยเสมอ

จากค่าความซับซ้อนที่ได้เราสามารถนำไปใช้ในการประเมินค่าใช้จ่ายซึ่งเป็นปัจจัยที่สำคัญในการพัฒนาโปรแกรม ถ้าโปรแกรมที่มีความซับซ้อนมากค่าใช้จ่ายในการพัฒนามักจะมากกว่าโปรแกรมที่มีความซับซ้อนน้อยกว่ารวมถึงสามารถนำไปประเมินระยะเวลาที่จะใช้ในการพัฒนาต่อไปด้วย

6.2 ข้อเสนอแนะ

เนื่องจากค่าความซับซ้อนที่ส่งผลต่อค่าใช้จ่าย และเวลาในการพัฒนาโปรแกรม แต่ไม่มีการกำหนดเกณฑ์ความสัมพันธ์ระหว่างค่าใช้จ่ายและระยะเวลาที่วัดได้ จึงกล่าวได้เพียงแต่ว่าโปรแกรมที่มีความซับซ้อนมากกว่าจะมีค่าใช้จ่ายและใช้เวลาในการพัฒนามากกว่าโปรแกรมที่มีความซับซ้อนน้อยกว่า ถ้ามีการกำหนดเกณฑ์ความสัมพันธ์ระหว่างค่าใช้จ่ายที่เหมาะสมกับค่าความซับซ้อนที่วัดได้ หรือกำหนดเกณฑ์ความสัมพันธ์ระหว่างเวลาที่เหมาะสมกับค่าความซับซ้อนที่วัดได้ การวัดความซับซ้อนจะนำไปใช้ประโยชน์ได้มากขึ้น นอกจากนี้การ

ศึกษาความสัมพันธ์ระหว่างความซับซ้อนกับค่าใช้จ่ายในการบำรุงรักษาก็จะเป็นอีกแนวทางหนึ่ง
ที่ควรทำการศึกษาต่อไป

6.3 แนวทางการวิจัยต่อ

1. ทดสอบตัววัด SDM กับโปรแกรมที่พัฒนาด้วยภาษาอื่น ๆ เช่น เดลไฟล์ วิซวลซี
2. กำหนดเกณฑ์มาตรฐานของโปรแกรมทางด้านธุรกิจ เช่น ระบบสินค้าคงคลัง ระบบบัญชี ระบบงานบุคคล ฯลฯ ควรมีค่าความซับซ้อนอยู่ในระดับใด
3. กำหนดค่าใช้จ่ายที่เหมาะสมในการพัฒนาโปรแกรมกับระดับของความซับซ้อน
4. กำหนดเวลาที่เหมาะสมในการพัฒนาโปรแกรมกับระดับของความซับซ้อน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Li H.F. and Cheung W.K. "An Empirical Study of Software Metrics." IEEE Trans. on Software Engineering., vol. SE-13, no. 6, June 1987. pp. 697-708.
- [2] ปราโมทย์ ลือนาม. "การพัฒนาโปรแกรมเพื่อใช้วัดความซับซ้อนของซอฟต์แวร์." วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย, จุฬาลงกรณ์มหาวิทยาลัย. 2535.
- [3] เอื้อน ปิ่นเงิน. "ความซับซ้อนของโปรแกรม." บิชีเนสคอมพิวเตอร์แมกะซีน, ปีที่ 4, มีนาคม 2535. หน้า 159-162.
- [4] Fenton N.E. and Shari L. P. **Software Metrics. A Rigorous and Practical Approach** 2nd Ed. FWS Publishing Company. 1997.
- [5] Harrison W. et.al "Applying Software Complexity Metrics to Program Maintenance." Computer, vol. 15, September 1982. pp. 65-79.
- [6] Basili V.R. et.al. "Metrics Analysis and Data Validation Across Fortran Project." IEEE Trans. on Software Engineering., vol. SE-9, no. 6, November 1983. pp. 652-663.
- [7] Fairley R.E. **Software Engineering Concepts.** McGraw-Hill, 1985.
- [8] McCabe T.J. "A Complexity Measure." Software Engineering Metrics I: Measure and Validations, vol. 1, 1993. pp. 22-51.
- [9] เอื้อน ปิ่นเงิน, นิภาพร ประภาศิริ. "การวัดความซับซ้อนของซอฟต์แวร์." สารสนเทศลาดกระบัง, ปีที่ 3, ฉบับที่ 1, มกราคม 2541. หน้า 42-55.
- [10] Woodward M.R. et.al. "A measure of control flow complexity in program text." IEEE Trans. on Software Engineering., vol. SE-5, no. 1, 1979. pp. 45-50.
- [11] Chen E.T. "Program complexity and programmer productivity." IEEE Trans. on Software Engineering., vol. SE-4, May 1978. pp. 187-197.
- [12] McClure C.L. **Structured Techniques for computing.** New Jersey : Prentice-Hall, Inc. 1985.
- [13] Henry S. and Kafura D. "Software structure metrics based on information flow." IEEE Ttrans. on Software Engineering., vol. SE-7, no. 5, September 1981. pp. 510-518.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [14] Shepperd M.J. and Ince D.C. "The Use of metrics in the early detection of design errors." Proceedings of the European Software Engineering Conference'90, 1990. pp. 67-85.
- [15] Oviedo E. "Control flow, data flow and program complexity." Software Engineering Metrics I: Measures and Validations, vol. 1, 1993. pp. 52-65.
- [16] สยามคอมพิวเตอร์. พัฒนาโปรแกรมด้วย Visual Basic. กรุงเทพมหานคร: ไทยเจริญการพิมพ์. 2538.
- [17] Gilb T. **Software Metrics**. Cambridge: Winthrop Publisher, Inc. 1977.
- [18] Pressman R. S. **Software Engineering : A practitioner's approach**. McGraw -Hill 4 ed., 1997.





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

คำสงวนภาษาวิชาวบลบลิค

A	CLnt	DefDate	EOF
Abs	Circle	DefDbl	Eqv
And	Clear	DefInt	Erase
AppActivate	CLng	DefLng	Exit Do
Array	Close	DefObj	Exit For
As	Command	DefSng	Exit Function
B	Compare	DefStr	Exit Property
Beep	Cos	DefVar	Exit Sub
Boolean	CreateObject	Dir	Exp
ByRef	CSng	Do	Explicit
Byte	CStr	DoEvents	F
ByVal	CurDir	Double	FileAttr
C	Currency	E	FileCopy
Call	Cvar	#Else	FileDateTime
Case	CVErr	Else	FileLen
Case Else	D	Elseif	Fix
CBool	Date	Empty	For
CByte	DateAdd	End	For Each
CCur	DateDiff	End Function	Format
CDate	DatePart	End If	FreeFile
CDbl	DateSerial	End Property	Function
CDecl	DateValue	End Select	FV
ChDir	DDB	End Sub	
ChDrive	Debug	End Type	G
Choose	Declare	End With	Get
Chr	DefCur	Environ	GetAttr

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GetObject	IsObject	Mid	Pmt
Global		MidB	Point
Go	K	Minute	PPmt
GoSub	Kill	MIRR	Preserve
GoTo		MkDir	Print
	L	Mod	Print #
H	LBound	Module	Private
Hex	Lcase	Month	Property Get
Hour	Left		Property Let
	Len	N	Property Set
I	LenB	Name	Pset
#If	Let	New	Public
If	Like	Next	Put
Imp	Line	Not	PV
Input	Line Input #	Nothing	
InputB	Load	Now	Q
Input #	LoadPicture	NPer	QBColor
Instr	Local	NPV	
InstrB	Loc	O	R
Int	Lock	Object	Raise
Integer	LOF	Oct	Randomize
IPmt	Log	On	Rate
IRR	Long	On Error	ReDim
Is	Loop	Open	Rem
IsArray	Lset	Option	Reset
IsDate	Ltrim	Option Base	Resume
IsEmpty		Optional	Return
IsError	M	Or	RGB
IsMissing	MacID		Right
IsNull	MacScript	P	Rmdir
IsNumeric	Me	ParamArray	Rnd

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RSet	SYD	W
Rtrim		Weekday
	T	Wend
S	Tab	While
SavePicture	Tan	Width #
Scale	Then	With
Second	Time	Write #
Seek	Timer	
Select Case	TimeSerial	X
SendKeys	TimeValue	Xor
Set	To	
SetAttr	Trim	Y
Sgn	Type	Year
Shared	TypeName	
Shell	TypeOf	
Sin	U	
Single	UBound	
SLN	UCase	
Space	Unload	
Spc	Unlock	
Sqr	Until	
Static		
Stop		
Str	V	
StrComp	Val	
StrConv	Variant	
String	VarType	
Sub		
Switch		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

ฟังก์ชันภาษาวิซวลเบสิก

A	DateDiff		Len
Abs	DatePart	H	LoadPicture
Array	DateSerial	Hex	Loc
Asc	DateValue	Hour	LOF
Atn	Day		Log
	DDB	I	LTrim
C	Dir	IIf	
CBool	DoEvents	IMEStatus	M
CByte		Input	MacID
CCur	E	InputBox	Mid
CDate	Environ	InStr	Minute
CDbl	EOF	Int	MIRR
Choose	Error	IPmt	Month
Chr	Exp	IRR	MsgBox
CInt	F	IsArray	N
CLng	FileAttr	IsDate	Now
Command	FileDateTime	IsEmpty	NPer
Cos	FileLen	IsError	NPV
CreateObject	Fix	IsMissing	
CSng	Format	IsNull	O
CStr	FreeFile	IsNumeric	Oct
CurDir	FV	IsObject	P
CVar			Partition
CVErr	G	L	Pmt
	GetAttr	LBound	PPmt
D	GetObject	LCase	PV
Date		Left	
DateAdd			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Q	TimeSerial
QBColor	TimeValue
	Trim
R	TypeName
Rate	
RGB	U
Right	UBound
Rnd	UCase
RTrim	
	V
S	Val
Second	VarType
Seek	
Sgn	W
Shell	Weekday
Sin	Y
SLN	Year
Space	
Spc	
Sqr	
Str	
StrComp	
StrConv	
String	
Switch	
SYD	
T	
Tab	
Tan	
Time	
Timer	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

ตารางแสดงผลการทดสอบการวัดความซับซ้อนของซอฟต์แวร์

ความหมายของตัวแปรต่างๆ ในตาราง

n1	คือ จำนวนโอเปอเรเตอร์ทั้งหมดที่ไม่ซ้ำกันในโปรแกรม (Unique Operators)
N1	คือ จำนวนโอเปอเรเตอร์ทั้งหมดในโปรแกรม (Total Operators)
n2	คือ จำนวนโอเปอเรนด์ทั้งหมดที่ไม่ซ้ำกันในโปรแกรม (Unique Operands)
N2	คือ จำนวนโอเปอเรนด์ทั้งหมดในโปรแกรม (Total Operands)
edge	คือ จำนวนเส้นเชื่อมระหว่างโหนดของกราฟควบคุมการไหล
node	คือ จำนวนโหนดทั้งหมดของกราฟควบคุมการไหล
df	คือ จำนวนการไหลของข้อมูลทั้งหมดในโปรแกรม
CFS	คือ จำนวนโครงสร้างเฉพาะของกราฟควบคุมการไหล
CFD	คือ จำนวนข้อมูลควบคุมการไหลของกราฟควบคุมการไหล
LOC	คือ จำนวนบรรทัดของโปรแกรม
Halstead	คือ ค่าความซับซ้อนจากตัววัดของ Halstead
McCabe	คือ ค่าความซับซ้อนจากตัววัดของ McCabe
Oviedo	คือ ค่าความซับซ้อนจากตัววัดของ Oviedo
SDM	คือ ค่าความซับซ้อนจากตัววัด SDM

ตารางแสดงผลการวัดค่าความซับซ้อนของระบบงานที่ 1

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
1	4	4	1	1	2	3	0	0	0	3	23.22	1	2	1
2	21	54	25	23	11	10	2	2	3	18	4108.54	3	13	6
3	31	202	32	114	48	38	102	11	13	60	104298.30	12	150	25
4	23	64	20	30	22	18	7	4	11	27	8798.69	6	29	16
5	21	54	14	23	11	10	2	2	3	18	6812.97	3	13	6
6	22	75	14	28	23	19	2	5	9	27	11715.05	6	25	15
7	13	25	9	14	2	3	0	0	0	8	1758.50	1	2	1
8	15	28	12	13	2	3	0	0	0	10	1583.97	1	2	1
9	9	11	3	3	2	3	0	0	0	3	225.85	1	2	1
10	7	28	10	15	2	3	0	0	0	10	922.74	1	2	1
11	30	80	19	46	25	21	16	5	6	31	25691.73	6	41	12
12	24	110	23	76	19	15	7	3	3	38	40966.78	6	26	7
13	22	148	31	88	17	14	22	4	5	37	42210.71	5	39	10
14	29	69	19	31	14	12	6	3	3	22	13212.85	4	20	7
15	12	44	17	28	13	10	10	1	1	22	3456.60	5	23	3
16	29	122	26	63	19	16	11	4	4	37	37578.28	5	30	9
17	35	120	24	55	29	23	23	7	7	33	41285.74	8	52	15
18	25	58	18	25	10	9	3	2	2	18	7819.10	3	13	5
19	15	13	5	8	5	5	3	1	5	5	1089.13	2	8	7
20	15	23	5	8	5	5	3	1	5	5	1607.76	2	8	7
21	15	29	7	11	7	7	2	1	2	9	2102.30	2	9	4
22	4	4	1	1	2	3	0	0	0	3	23.22	1	2	1
23	11	27	16	18	2	3	0	0	0	10	1323.94	1	2	1
24	9	11	3	3	2	3	0	0	0	3	225.85	1	2	1
25	15	23	5	8	5	5	3	1	4	5	1607.76	2	8	6
26	20	29	7	9	7	7	0	1	1	13	2323.10	2	7	3
27	29	127	32	59	19	16	11	4	10	36	29491.15	5	30	15
28	38	455	71	188	85	68	38	18	48	109	218945.62	19	123	67
29	4	4	1	1	2	3	0	0	0	3	23.22	1	2	1
30	13	15	2	2	6	6	0	1	1	7	431.71	2	6	3
31	17	47	16	30	6	6	1	1	1	19	6190.42	2	7	3
32	6	79	25	42	2	3	0	0	0	23	3021.27	1	2	1
33	19	39	9	17	14	12	3	3	4	16	4830.86	4	17	8
34	23	47	18	24	10	9	0	2	2	18	5832.59	3	10	5
35	9	24	10	13	2	3	0	0	0	11	919.46	1	2	1
36	19	44	8	17	7	7	3	1	1	9	5855.35	2	10	3
37	19	75	14	33	7	7	3	1	1	11	12199.51	2	10	3
38	19	45	8	17	7	7	3	1	1	9	5951.34	2	10	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
39	7	35	16	30	2	3	1	0	0	17	1929.58	1	3	1
40	24	187	46	94	15	13	4	2	5	41	42234.49	4	19	8
41	27	88	16	35	22	18	19	5	8	29	19710.06	6	41	14
42	14	69	23	44	2	3	0	0	0	23	7883.04	1	2	1
43	15	43	17	28	9	8	3	2	3	18	4385.29	3	12	6
44	9	25	11	13	5	5	0	1	1	10	873.42	2	5	3
45	9	20	9	10	5	5	0	1	1	9	625.49	2	5	3
46	15	22	5	8	5	5	3	1	4	5	1555.89	2	8	6
47	12	25	9	12	6	6	2	1	1	7	1300.13	2	8	3
48	4	4	1	1	2	3	0	0	0	3	23.22	1	2	1
49	15	23	5	8	5	5	3	1	4	5	1607.76	2	8	6
50	12	25	9	12	6	6	2	1	1	7	1300.13	2	8	3
51	7	39	11	24	2	3	0	0	0	10	2006.11	1	2	1
52	7	9	3	4	2	3	0	0	0	3	201.53	1	2	1
53	29	94	21	29	32	26	4	7	8	39	13900.41	8	36	16
54	9	13	5	5	2	3	0	0	0	4	308.40	1	2	1
55	8	10	3	3	2	3	0	0	0	4	179.89	1	2	1
56	25	67	15	25	15	13	2	3	5	21	10200.36	4	17	9
57	31	203	36	111	49	37	71	10	12	62	91031.27	14	120	23
58	32	227	41	108	71	56	28	13	31	81	87394.28	17	99	45
59	30	96	21	29	32	26	4	7	8	39	14687.53	8	36	16
60	26	68	14	25	15	13	2	3	5	21	11489.66	4	17	9
61	27	124	25	54	36	28	9	7	8	45	29588.02	10	45	16
62	20	44	13	21	7	7	0	1	2	14	5296.61	2	7	4
63	23	88	39	58	10	9	1	2	2	29	14867.48	3	11	5
64	13	13	3	3	2	3	0	0	0	6	416.00	1	2	1
65	7	32	11	17	2	3	0	0	0	11	1105.22	1	2	1
66	30	84	20	48	25	21	17	5	6	31	26819.60	6	42	12
67	22	28	7	13	11	10	3	2	2	14	4068.91	3	14	5
68	30	92	24	41	20	17	3	4	4	32	19613.38	5	23	9
69	25	230	37	129	32	25	78	8	13	51	93157.02	9	110	22
70	20	169	34	96	22	18	38	5	6	40	43060.10	6	60	12
71	30	186	39	104	23	19	17	5	6	49	70858.88	6	40	12
72	35	123	23	58	29	23	28	7	9	33	46791.26	8	57	17
73	26	59	18	26	10	9	4	2	2	18	8713.86	3	14	5
74	26	77	22	37	10	9	3	2	2	21	13920.27	3	13	5
75	26	80	23	38	10	9	4	2	2	21	14230.12	3	14	5
76	23	74	24	44	19	16	2	4	4	27	13818.89	5	21	9
77	15	23	5	8	5	5	3	1	4	5	1607.76	2	8	6
78	14	39	14	20	10	9	2	2	3	13	2836.34	3	12	6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2.	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
79	12	25	9	12	6	6	2	1	1	7	1300.13	2	8	3
80	9	55	12	42	11	9	0	1	1	21	6710.36	4	11	3
81	15	23	5	8	5	5	3	1	4	5	1607.76	2	8	6
82	11	13	3	3	5	5	0	1	1	5	335.05	2	5	3
83	15	29	7	11	10	9	0	2	3	13	2102.30	3	10	6
84	15	28	10	15	7	7	2	1	2	9	2246.47	2	9	4
85	14	21	9	10	11	9	1	1	1	13	1090.68	4	12	3
86	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
87	7	7	1	1	2	3	0	0	0	3	84.00	1	2	1
88	29	94	21	29	32	26	4	7	8	39	13900.41	8	36	16
89	4	4	1	1	2	3	0	0	0	3	23.22	1	2	1
90	22	54	13	22	11	10	2	2	3	18	7256.75	3	13	6
91	31	216	32	114	48	38	102	11	13	59	108919.12	12	150	25
92	30	96	21	29	32	26	4	7	8	39	14687.53	8	36	16
93	24	80	25	41	26	20	11	4	11	31	13370.20	8	37	16
94	5	6	2	2	2	3	0	0	0	3	56.15	1	2	1
95	20	50	13	20	11	10	2	2	2	18	5432.42	3	13	5
96	24	90	18	38	23	19	2	5	9	33	17485.49	6	25	15
97	13	19	6	7	2	3	0	0	0	6	837.55	1	2	1
98	21	74	29	39	11	10	1	2	2	25	9005.55	3	12	5
99	10	13	3	3	2	3	0	0	0	4	296.04	1	2	1
100	10	50	16	31	9	8	3	2	3	18	3688.38	3	12	6
101	30	84	20	48	25	21	16	5	8	31	26819.60	6	41	14
102	24	110	23	77	19	15	7	3	3	38	41728.97	6	26	7
103	30	92	24	41	20	17	3	4	4	32	19613.38	5	23	9
104	22	142	31	86	17	14	22	4	5	37	39853.02	5	39	10
105	28	68	19	31	14	12	3	3	3	22	12560.97	4	17	7
106	34	247	38	119	22	18	8	5	7	39	120219.04	6	30	13
107	12	44	17	28	13	10	10	1	1	22	3456.60	5	23	3
108	29	134	28	69	19	16	11	4	4	37	42309.60	5	30	9
109	35	122	24	56	29	23	23	7	9	33	42757.01	8	52	17
110	25	58	18	25	10	9	3	2	2	18	7819.10	3	13	5
111	15	23	5	8	5	5	3	1	5	5	1607.76	2	8	7
112	15	23	5	8	5	5	3	1	5	5	1607.76	2	8	7
113	15	28	10	15	10	9	0	2	3	13	2246.47	3	10	6
114	19	38	10	15	10	9	3	2	3	12	3668.99	3	13	6
115	15	29	7	11	7	7	2	1	2	10	2102.30	2	9	4
รวม										2393	1960845.69	444	2459	881

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงผลการวัดค่าความซับซ้อนของระบบงานที่ 2

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
1	5	5	0	0	2	3	0	0	0	3	0.00	7	5	6
2	5	5	1	1	2	3	0	0	0	3	38.77	6	6	6
3	33	219	48	153	28	23	10	5	8	68	124038.37	173	372	52
4	16	40	10	34	13	11	3	3	3	17	9461.05	32	74	25
5	19	99	15	58	33	27	1	6	6	44	29340.08	86	157	61
6	19	164	24	139	23	19	1	4	4	64	90462.96	142	303	43
7	11	17	6	7	2	3	0	0	0	6	629.47	13	24	6
8	14	70	24	35	2	3	0	0	0	21	5625.12	48	105	6
9	37	336	40	169	50	40	51	10	12	111	247362.99	298	505	91
10	5	5	0	0	2	3	0	0	0	3	0.00	7	5	6
11	21	57	16	23	14	12	1	2	4	18	6290.41	43	80	27
12	21	63	18	27	14	12	-1	2	4	-20	7492.06	47	90	27
13	20	64	26	40	7	7	1	1	2	20	8837.70	40	104	15
14	24	92	35	50	15	13	4	3	4	27	14320.03	59	142	29
15	13	23	11	14	2	3	0	0	0	7	1403.42	14	37	6
16	21	47	20	27	10	9	3	2	2	16	5619.80	29	74	20
17	9	47	20	26	2	3	0	0	0	15	2074.60	29	73	6
18	7	8	2	2	2	3	0	0	0	5	110.95	8	10	6
19	27	179	35	101	27	21	2	5	5	60	64948.37	146	280	49
20	13	27	9	16	6	6	0	1	1	10	2215.84	20	43	13
21	18	63	21	46	13	11	2	3	3	24	11357.57	44	109	25
22	13	27	10	16	6	6	0	1	1	10	2022.94	19	43	13
23	5	5	0	0	2	3	0	0	0	3	0.00	7	5	6
24	23	188	35	91	29	23	13	6	11	54	48867.86	155	279	53
25	27	806	142	514	149	111	49	31	41	254	477381.74	666	1320	261
26	31	239	39	144	67	51	14	16	21	75	134350.11	202	383	119
27	12	21	9	9	8	7	0	2	2	8	790.62	14	30	16
28	12	20	8	10	5	5	1	1	1	7	972.43	14	30	11
29	8	8	3	3	2	3	0	0	0	3	152.21	7	11	6
30	31	442	97	267	31	25	1	5	9	124	211746.14	347	709	57
31	29	142	38	79	19	16	5	4	4	38	40412.21	106	221	36
32	18	36	14	20	8	7	1	2	2	10	3600.00	24	56	16
33	10	14	6	9	2	3	0	0	0	6	690.00	10	23	6
34	5	5	0	0	2	3	0	0	0	3	0.00	7	5	6
รวม										1157	1552615.83	2869	5713	1131

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงผลการวัดค่าความซับซ้อนของระบบงานที่ 3

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
1	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
2	25	261	22	93	15	13	0	3	0	37	103902.37	4	15	4
3	8	24	12	12	2	3	0	0	3	9	622.36	1	2	4
4	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
5	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
6	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
7	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
8	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
9	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
10	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
11	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
12	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
13	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
14	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
15	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
16	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
17	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
18	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
19	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
20	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
21	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
22	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
23	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
24	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
25	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
26	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
27	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
28	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
29	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
30	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
31	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
32	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
33	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
34	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
35	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
36	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
37	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
38	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
39	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
40	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
41	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
42	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
43	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
44	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
45	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
46	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
47	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
48	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
49	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
50	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
51	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
52	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
53	20	548	57	280	140	112	182	27	40	184	254893.30	30	322	68
54	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
55	16	387	52	168	99	72	292	13	28	127	87322.31	29	391	42
56	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
57	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
58	20	400	48	216	102	78	72	17	25	137	168744.47	26	174	43
59	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
60	16	283	43	126	82	59	144	8	24	94	56401.14	25	226	33
61	20	73	12	25	15	13	0	3	3	20	10208.33	4	15	7
62	22	68	19	31	9	8	0	2	2	20	9519.24	3	9	5
63	20	165	29	77	37	31	12	7	13	53	36077.41	8	49	21
64	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
65	12	84	15	28	26	21	0	6	6	23	5964.53	7	26	13
66	6	6	2	2	2	3	0	0	0	3	72.00	1	2	1
67	18	37	8	11	8	7	0	2	2	10	2792.06	3	8	5
68	8	8	2	2	2	3	0	0	0	3	132.88	1	2	1
69	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
70	20	147	23	82	36	27	7	5	7	51	44301.91	11	43	13
71	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
72	16	96	24	45	31	23	18	2	9	33	11255.88	10	49	12
73	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
74	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
75	20	78	22	36	18	15	2	3	4	25	10059.12	5	20	8
76	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
77	15	41	13	18	12	10	3	1	3	15	2945.43	4	15	5
78	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
79	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
80	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
81	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
82	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
83	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
84	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
85	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
86	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
87	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
88	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
89	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
90	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
91	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
92	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
93	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
94	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
95	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
96	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
97	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
98	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
99	15	52	13	19	10	9	0	2	2	15	3741.42	3	10	5
100	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
101	9	13	5	5	2	3	0	0	0	5	308.40	1	2	1
102	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
103	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
104	20	400	46	208	90	72	72	17	25	119	166173.53	20	162	43
105	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
106	16	281	41	123	64	47	123	8	18	82	56555.70	19	187	27
107	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
108	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
109	20	462	50	234	118	95	132	23	34	155	199647.91	25	250	58
110	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
111	16	183	33	93	82	60	209	11	23	107	34937.79	24	291	35
112	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
113	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
114	20	72	20	34	18	15	2	3	4	25	9590.11	5	20	8
115	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
116	16	45	15	21	12	10	4	1	3	16	3662.14	4	16	5
117	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
118	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
119	14	25	7	8	6	6	0	1	1	9	1159.57	2	6	3
120	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
121	9	10	4	4	2	3	0	0	0	4	233.13	1	2	1
122	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
123	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
124	14	25	7	8	6	6	0	1	1	9	1159.57	2	6	3
125	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
126	9	10	4	4	2	3	0	0	0	4	233.13	1	2	1
127	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
128	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
129	14	25	7	8	6	6	0	1	1	9	1159.57	2	6	3
130	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
131	9	10	4	4	2	3	0	0	0	4	233.13	1	2	1
132	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
133	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
134	21	380	55	210	96	72	56	15	22	131	147786.21	26	152	38
135	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
136	17	261	24	120	81	58	139	7	24	88	86752.16	25	220	32
137	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
138	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
139	20	401	49	216	102	78	72	17	25	137	166141.89	26	174	43
140	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
141	7	7	3	3	2	3	0	0	0	3	116.27	1	2	1
142	14	18	5	5	6	6	0	1	1	8	683.92	2	6	3
143	12	22	10	13	5	5	1	1	1	9	1217.42	2	6	3
144	14	93	27	48	5	5	1	1	1	28	9400.72	2	6	3
145	23	60	18	27	13	11	1	3	3	21	8040.35	4	14	7
146	6	21	10	14	2	3	0	0	0	9	588.00	1	2	1
147	10	461	168	429	59	42	171	18	18	197	84949.70	19	230	37
148	8	17	7	7	2	3	0	0	0	4	375.06	1	2	1
149	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
150	21	95	19	39	21	17	3	5	5	27	15370.01	6	24	11
151	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
152	8	10	4	4	2	3	0	0	0	6	200.76	1	2	1
153	8	10	4	4	2	3	0	0	0	6	200.76	1	2	1
154	8	10	4	4	2	3	0	0	0	6	200.76	1	2	1
155	8	10	4	4	2	3	0	0	0	6	200.76	1	2	1
156	6	6	2	2	2	3	0	0	0	3	72.00	1	2	1
157	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
158	43	4350	444	2383	1435	1039	32719	394	696	1382	6936358.44	398	34154	1091
159	6	12	5	8	2	3	0	0	0	6	332.11	1	2	1
160	26	160	22	117	52	40	57	10	20	53	106956.35	14	109	31
161	11	13	6	8	5	5	2	1	1	6	629.47	2	7	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
162	11	13	6	8	5	5	2	1	1	6	629.47	2	7	3
163	6	14	6	10	2	3	0	0	0	7	430.20	1	2	1
164	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
165	19	412	61	264	122	93	669	30	54	143	175708.97	31	791	85
166	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
167	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
168	19	344	53	220	102	78	470	25	45	120	137223.79	26	572	71
169	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
170	19	412	61	264	122	93	669	30	54	143	175708.97	31	791	85
171	19	126	27	76	36	28	61	8	15	43	29836.24	10	97	24
172	19	158	32	95	42	32	78	9	17	50	40474.97	12	120	27
173	19	208	38	129	54	40	112	11	21	64	63393.31	16	166	33
174	19	182	35	112	48	36	95	10	19	57	51434.88	14	143	30
175	19	338	53	214	84	60	197	16	31	99	130641.29	26	281	48
176	19	208	38	129	54	40	112	11	21	64	63393.31	16	166	33
177	19	208	38	129	54	40	112	11	21	64	63393.31	16	166	33
178	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
179	19	684	93	440	202	153	1815	50	90	235	343904.64	51	2017	141
180	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
181	20	20	1	1	2	3	0	0	0	16	922.39	1	2	1
182	19	103	26	61	30	24	44	7	13	36	20074.41	8	74	21
183	19	103	26	61	30	24	44	7	13	36	20074.41	8	74	21
184	19	103	26	61	30	24	44	7	13	36	20074.41	8	74	21
185	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
186	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
187	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
188	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
189	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
190	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
191	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
192	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
193	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
194	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
195	19	130	27	76	36	28	61	8	15	43	30427.05	10	97	24
196	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
197	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
198	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
199	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
200	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
201	7	12	6	6	2	3	0	0	0	4	233.13	1	2	1
202	13	18	6	6	2	3	0	0	0	5	662.68	1	2	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
203	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
204	13	30	14	18	5	6	1	1	1	12	1907.39	1	6	3
205	13	30	14	18	5	6	1	1	1	12	1907.39	1	6	3
206	13	31	15	19	5	6	1	1	1	12	1979.03	1	6	3
207	14	31	15	19	5	6	1	1	1	12	2153.70	1	6	3
208	22	66	24	34	9	9	1	2	1	24	8607.55	2	10	4
209	14	30	17	18	5	6	1	1	1	15	1762.53	1	6	3
210	14	30	17	18	5	6	1	1	1	15	1762.53	1	6	3
211	7	10	4	4	2	3	0	0	0	4	169.51	1	2	1
212	7	10	4	4	2	3	0	0	0	4	169.51	1	2	1
213	13	30	14	18	5	6	1	1	1	12	1907.39	1	6	3
214	7	13	6	6	2	3	0	0	0	4	246.08	1	2	1
215	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
216	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
217	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
218	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
219	7	10	4	4	2	3	0	0	0	4	169.51	1	2	1
220	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
221	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
222	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
223	6	6	1	1	2	3	0	0	0	3	58.95	1	2	1
224	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
225	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
226	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
227	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
228	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
229	7	34	10	18	2	3	0	0	0	9	1339.05	1	2	1
230	13	76	26	39	2	3	0	0	0	15	5926.26	1	2	1
231	12	17	4	5	6	6	0	1	1	7	660.00	2	6	3
232	8	318	82	158	2	3	0	0	0	81	23816.55	1	2	1
233	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
234	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
235	7	34	10	18	2	3	0	0	0	9	1339.05	1	2	1
236	13	76	26	39	2	3	0	0	0	15	5926.26	1	2	1
237	12	17	4	5	6	6	0	1	1	7	660.00	2	6	3
238	8	366	94	182	2	3	0	0	0	93	28318.34	1	2	1
239	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
240	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
241	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
242	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
243	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
244	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
245	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
246	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
247	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
248	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
249	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
250	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
251	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
252	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
253	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
254	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
255	8	8	2	2	2	3	0	0	0	3	132.88	1	2	1
256	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
257	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
258	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
259	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
260	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
261	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
262	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
263	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
264	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
265	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
266	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
267	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
268	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
269	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
270	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
271	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
272	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
273	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
274	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
275	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
276	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
277	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
278	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
279	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
280	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
281	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
282	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
283	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
284	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
285	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
286	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
287	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
288	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
289	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
290	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
291	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
292	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
293	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
294	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
295	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
296	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
297	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
298	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
299	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
300	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
301	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
302	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
303	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
304	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
305	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
306	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
307	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
308	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
309	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
310	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
311	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
312	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
313	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
314	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
315	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
316	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
317	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
318	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
319	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
320	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
321	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
322	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
323	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
324	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
325	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
326	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
327	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
328	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
329	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
330	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
331	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
332	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
333	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
334	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
335	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
336	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
337	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
338	8	8	2	2	2	3	0	0	0	3	132.88	1	2	1
339	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
340	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
341	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
342	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
343	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
344	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
345	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
346	9	33	13	17	2	3	0	0	0	9	1312.10	1	2	1
347	9	33	13	17	2	3	0	0	0	9	1312.10	1	2	1
348	9	33	13	17	2	3	0	0	0	9	1312.10	1	2	1
349	9	33	13	17	2	3	0	0	0	9	1312.10	1	2	1
350	9	33	13	17	2	3	0	0	0	9	1312.10	1	2	1
351	9	33	13	17	2	3	0	0	0	9	1312.10	1	2	1
352	9	33	13	17	2	3	0	0	0	9	1312.10	1	2	1
353	9	33	13	17	2	3	0	0	0	9	1312.10	1	2	1
354	9	33	13	17	2	3	0	0	0	9	1312.10	1	2	1
355	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
356	8	8	2	2	2	3	0	0	0	3	132.88	1	2	1
357	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
358	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
359	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
360	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
361	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
362	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
363	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
364	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
365	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
366	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
367	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
368	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
369	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
370	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
371	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
372	8	8	2	2	2	3	0	0	0	3	132.88	1	2	1
373	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
374	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
375	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
376	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
377	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
378	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
379	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
380	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
381	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
382	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
383	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
384	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
385	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
386	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
387	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
388	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
389	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
390	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
391	9	9	3	3	2	3	0	0	0	4	193.59	1	2	1
392	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
393	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
394	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
395	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
396	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
397	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
398	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
399	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
400	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
401	7	30	10	16	2	3	0	0	0	8	1052.93	1	2	1
402	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
403	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
404	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
405	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
406	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
407	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
408	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
409	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
410	8	8	2	2	2	3	0	0	0	3	132.88	1	2	1
411	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
412	6	6	1	1	2	3	0	0	0	3	58.95	1	2	1
413	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
414	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
415	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
416	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
417	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
418	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
419	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
420	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
421	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
422	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
423	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
424	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
425	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
426	9	33	13	17	2	3	1	0	0	8	1312.10	1	3	1
427	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
428	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
429	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
430	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
431	9	33	13	17	2	3	0	0	0	8	1312.10	1	2	1
432	11	16	4	5	5	5	0	1	1	6	564.06	2	5	3
433	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
434	8	24	12	14	2	3	0	0	0	9	766.42	1	2	1
435	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
436	20	47	16	16	9	8	0	2	2	13	3257.05	3	9	5
437	11	39	11	30	6	6	0	1	1	19	4615.51	2	6	3
438	25	93	24	38	19	16	0	4	5	28	14557.31	5	19	10
439	13	44	18	27	5	5	0	1	2	17	3429.54	2	5	4
440	32	132	27	48	29	24	1	6	7	37	30119.13	7	30	14
441	14	51	18	27	7	7	0	1	2	17	4095.00	2	7	4
442	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
443	6	7	1	1	2	3	0	0	0	3	67.38	1	2	1
444	6	7	1	1	2	3	0	0	0	3	67.38	1	2	1
445	12	23	1	4	9	8	0	2	2	8	2397.88	3	9	5
446	11	16	1	3	6	6	0	1	1	6	1123.89	2	6	3
447	23	157	40	69	12	11	0	2	1	40	26797.79	3	12	4
448	6	10	3	4	2	3	0	0	0	4	177.52	1	2	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
449	9	21	8	10	2	3	0	0	0	7	712.75	1	2	1
450	6	10	3	4	2	3	0	0	0	4	177.52	1	2	1
451	22	70	19	29	9	8	0	2	2	21	8905.10	3	9	5
452	8	8	2	2	2	3	0	0	0	3	132.88	1	2	1
453	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
454	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
455	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
456	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
457	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
458	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
459	22	59	18	24	9	8	0	2	2	17	6478.56	3	9	5
460	7	35	12	21	2	3	0	0	0	13	1457.04	1	2	1
461	6	6	1	1	2	3	0	0	0	3	58.95	1	2	1
462	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
463	7	35	12	21	2	3	0	0	0	13	1457.04	1	2	1
464	6	6	1	1	2	3	0	0	0	3	58.95	1	2	1
465	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
466	7	42	17	26	2	3	0	0	0	14	1668.93	1	2	1
467	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
468	6	6	1	1	2	3	0	0	0	3	58.95	1	2	1
469	7	7	2	2	2	3	0	0	0	4	99.85	1	2	1
470	21	62	15	22	13	11	0	3	3	18	6687.81	4	13	7
471	25	52	16	23	9	8	3	2	2	16	7220.14	3	12	5
472	8	10	3	3	2	3	0	0	0	5	179.89	1	2	1
473	24	50	13	21	9	8	3	2	0	15	7169.81	3	12	3
474	7	48	23	43	2	3	0	0	0	24	2921.84	1	2	1
475	13	17	3	3	5	5	0	1	2	6	520.00	2	5	4
476	7	7	1	1	2	3	0	0	0	4	84.00	1	2	1
477	13	17	3	3	5	5	1	1	2	6	520.00	2	6	4
478	13	17	3	3	5	5	1	1	2	6	520.00	2	6	4
479	13	17	3	3	5	5	1	1	2	6	520.00	2	6	4
480	13	17	3	3	5	5	0	1	2	6	520.00	2	5	4
481	13	17	3	3	5	5	0	1	2	6	520.00	2	5	4
482	13	17	3	3	5	5	0	1	2	6	520.00	2	5	4
483	12	72	14	39	56	43	0	14	26	43	8720.66	15	56	41
484	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
485	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
486	26	54	15	23	15	13	0	3	3	23	8223.13	4	15	7
487	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
488	26	54	15	23	15	13	0	3	3	23	8223.13	4	15	7
489	18	28	5	9	9	8	0	2	2	11	2711.42	3	9	5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
490	18	4203	199	1525	12	11	1	2	2	396	3066272.97	3	13	5
491	17	2123	199	769	12	11	1	2	2	228	736658.12	3	13	5
492	22	288	20	87	22	19	1	4	6	40	96758.40	5	23	11
493	18	37	8	11	8	7	0	2	2	10	2792.06	3	8	5
494	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
495	22	67	17	27	9	8	0	2	2	17	8679.87	3	9	5
496	2	7	2	2	2	3	0	0	0	3	18.00	1	2	1
497	20	162	33	71	21	18	2	4	4	42	28714.24	5	23	9
498	15	56	21	29	10	9	2	2	2	17	4551.38	3	12	5
499	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
500	15	23	7	8	7	7	0	1	1	8	1184.93	2	7	3
501	21	73	17	28	9	8	0	2	2	19	9166.59	3	9	5
502	23	76	11	24	11	9	0	3	3	18	12764.91	4	11	7
503	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
504	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
505	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
506	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
507	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
508	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
509	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
510	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
511	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
512	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
513	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
514	8	19	8	8	2	3	0	0	0	5	432.00	1	2	1
515	18	37	8	11	8	7	0	2	2	10	2792.06	3	8	5
516	24	50	8	15	9	8	0	2	2	16	7312.50	3	9	5
517	24	50	8	15	9	8	0	2	2	16	7312.50	3	9	5
518	24	50	8	15	9	8	0	2	2	16	7312.50	3	9	5
519	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
520	20	30	6	10	9	8	0	2	2	14	3133.63	3	9	5
521	8	8	2	2	2	3	0	0	0	3	132.88	1	2	1
522	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
523	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
524	21	54	10	17	13	11	0	3	4	15	6278.70	4	13	8
525	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
526	19	41	8	13	9	8	0	2	2	12	3963.79	3	9	5
รวม										9466	15214076.55	1821	45945	3583

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงผลการวัดค่าความซับซ้อนของระบบงานที่ 4

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
1	5	5	3	3	2	3	0	0	0	4	60.00	1	2	1
2	22	75	17	30	10	9	1	2	2	11	10772.89	3	11	5
3	6	6	3	3	2	3	0	0	0	4	85.59	1	2	1
4	6	9	4	4	2	3	0	0	0	4	129.56	1	2	1
5	7	16	8	11	2	3	0	0	0	6	507.65	1	2	1
6	14	19	6	7	5	5	0	1	1	7	917.69	2	5	3
7	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
8	31	121	26	62	35	27	8	9	19	27	39453.44	10	43	29
9	21	93	27	43	9	8	0	2	2	15	12701.45	3	9	5
10	35	134	42	66	25	21	4	5	5	40	34467.33	6	29	11
11	7	7	2	2	2	3	0	0	0	3	99.85	1	2	1
12	25	45	15	23	15	13	8	3	4	14	6936.25	4	23	8
13	9	14	10	11	6	6	0	1	1	8	525.68	2	6	3
14	19	44	15	20	13	11	0	3	2	15	4124.24	4	13	6
15	17	36	15	18	9	8	0	2	2	10	2754.00	3	9	5
16	7	28	12	15	2	3	0	0	0	10	799.14	1	2	1
17	9	12	5	6	6	6	0	1	1	6	370.07	2	6	3
18	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
19	6	7	3	4	2	3	0	0	0	5	139.48	1	2	1
20	10	43	14	25	6	6	0	1	1	9	2783.73	2	6	3
21	17	70	14	30	14	12	6	3	4	14	9023.71	4	20	8
22	11	48	14	26	9	8	6	2	2	11	3510.09	3	15	5
23	21	50	17	22	13	11	0	3	3	15	5134.33	4	13	7
24	11	27	15	17	6	6	0	1	1	9	1289.17	2	6	3
25	7	28	11	14	2	3	0	0	0	10	780.16	1	2	1
26	9	12	5	6	6	6	0	1	1	6	370.07	2	6	3
27	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
28	6	7	3	4	2	3	0	0	0	5	139.48	1	2	1
29	10	51	17	31	6	6	0	1	1	11	3554.98	2	6	3
30	19	156	25	60	26	21	12	6	7	26	26886.61	7	38	14
31	13	103	22	49	18	14	21	5	5	14	11287.22	6	39	11
32	6	6	3	3	2	3	0	0	0	3	85.59	1	2	1
33	5	7	3	4	2	3	0	0	0	4	110.00	1	2	1
34	14	42	13	19	11	10	0	2	2	14	2967.42	3	11	5
35	17	46	14	25	6	6	3	1	1	16	5339.03	2	9	3
36	27	166	33	81	37	30	24	8	13	48	48346.02	9	61	22
37	4	4	1	1	2	3	0	0	0	3	23.22	1	2	1
38	28	83	20	34	17	14	6	4	4	14	15551.89	5	23	9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
39	8	11	7	7	2	3	0	0	0	6	281.30	1	2	1
40	7	7	1	1	2	3	0	0	0	3	84.00	1	2	1
41	6	9	4	4	2	3	0	0	0	4	129.56	1	2	1
42	6	9	4	4	2	3	0	0	0	4	129.56	1	2	1
43	6	9	4	4	2	3	0	0	0	4	129.56	1	2	1
44	7	9	5	5	2	3	0	0	0	3	175.66	1	2	1
45	7	20	7	11	2	3	0	0	0	8	649.15	1	2	1
46	18	72	18	29	18	15	0	4	6	23	7571.36	5	18	11
47	22	83	33	49	11	10	3	2	2	27	12464.61	3	14	5
48	7	9	5	5	2	3	0	0	0	3	175.66	1	2	1
49	7	23	8	13	2	3	0	0	0	9	799.94	1	2	1
50	8	32	17	17	2	3	0	0	0	10	910.20	1	2	1
51	23	120	39	72	15	13	9	3	3	37	24271.14	4	24	7
52	6	6	3	3	2	3	0	0	0	3	85.59	1	2	1
53	8	26	8	13	2	3	0	0	0	6	1014.00	1	2	1
54	8	32	17	17	2	3	0	0	0	10	910.20	1	2	1
55	23	120	42	72	15	13	9	3	3	37	22795.52	4	24	7
56	11	23	13	17	13	11	0	2	3	12	1319.06	4	13	6
57	19	47	17	24	20	16	0	4	5	19	4922.99	6	20	10
58	12	31	14	18	9	8	0	1	2	9	1776.77	3	9	4
59	18	37	16	21	16	13	0	3	4	14	3485.55	5	16	8
60	7	26	12	15	2	3	0	0	0	10	761.97	1	2	1
61	11	38	14	21	9	8	0	1	2	14	2260.40	3	9	4
62	11	21	9	13	13	11	0	2	3	10	1167.40	4	13	6
63	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
64	12	28	13	18	9	8	2	1	2	12	1774.67	3	11	4
65	11	189	37	115	6	6	0	1	1	31	29023.69	2	6	3
66	9	11	4	5	2	3	1	0	0	4	333.04	1	3	1
67	15	60	17	45	26	20	3	3	7	23	10422.79	8	29	11
68	16	344	37	123	55	42	20	13	14	49	71138.91	15	75	28
69	17	241	44	124	56	39	151	14	15	32	51854.86	19	207	30
70	6	6	3	3	2	3	0	0	0	3	85.59	1	2	1
71	29	122	27	57	30	24	8	7	7	25	31820.76	8	38	15
72	10	16	8	9	2	3	0	0	0	6	586.40	1	2	1
73	6	9	4	4	2	3	0	0	0	4	129.56	1	2	1
74	6	9	4	4	2	3	0	0	0	4	129.56	1	2	1
75	19	41	17	22	11	10	1	2	3	11	4004.26	3	12	6
76	24	355	54	184	36	29	55	8	8	50	138524.68	9	91	17
77	22	368	32	160	55	42	47	14	18	47	167121.93	15	102	33
78	4	4	1	1	2	3	0	0	0	3	23.22	1	2	1
79	13	39	10	27	8	7	1	1	1	8	5239.64	3	9	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
80	12	16	7	10	7	7	1	1	1	7	946.68	2	8	3
81	12	29	12	16	11	9	1	1	1	9	1650.59	4	12	3
82	12	21	5	9	7	7	1	1	1	9	1324.34	2	8	3
83	21	51	11	19	12	10	6	3	4	17	6347.73	4	18	8
84	8	12	4	4	2	3	0	0	0	5	229.44	1	2	1
85	7	9	6	6	2	3	0	0	0	6	194.27	1	2	1
86	8	13	7	7	2	3	0	0	0	7	312.55	1	2	1
87	28	92	26	47	22	18	3	4	5	36	20244.37	6	25	10
88	16	34	9	15	6	6	3	1	1	10	3033.99	2	9	3
89	4	4	0	0	2	3	0	0	0	3	0.00	1	2	1
90	11	19	4	6	2	3	0	0	0	5	805.80	1	2	1
91	15	40	14	25	12	10	0	1	3	18	4229.05	4	12	5
92	11	51	17	29	6	6	0	1	1	11	3608.34	2	6	3
93	25	78	18	43	12	11	13	2	4	18	19606.15	3	25	7
94	14	30	10	14	9	8	0	1	2	10	1977.04	3	9	4
95	22	90	26	53	23	19	7	4	7	26	17908.18	6	30	12
96	10	55	17	29	6	6	0	1	1	15	3406.74	2	6	3
97	22	184	21	73	42	31	60	12	15	54	53324.94	13	102	28
98	8	12	4	4	2	3	0	0	0	5	229.44	1	2	1
99	8	9	6	6	2	3	0	0	0	6	228.44	1	2	1
100	8	9	6	6	2	3	0	0	0	6	228.44	1	2	1
101	8	13	6	7	2	3	0	0	0	7	355.35	1	2	1
102	28	129	32	69	26	21	5	4	7	41	35306.22	7	31	12
103	11	49	11	18	2	3	0	0	0	8	2689.04	1	2	1
104	15	72	19	51	15	12	0	1	4	27	12597.50	5	15	6
105	11	167	54	107	6	6	0	1	1	25	17983.35	2	6	3
106	9	24	9	10	2	3	0	0	0	6	708.89	1	2	1
107	22	257	27	101	62	45	126	18	24	76	82710.50	19	188	43
108	8	12	4	4	2	3	0	0	0	5	229.44	1	2	1
109	7	9	6	6	2	3	0	0	0	6	194.27	1	2	1
110	7	9	6	6	2	3	0	0	0	6	194.27	1	2	1
111	8	13	6	7	2	3	0	0	0	7	355.35	1	2	1
112	28	133	34	70	26	21	5	4	7	42	34839.05	7	31	12
113	16	78	26	49	7	7	3	1	1	25	10325.04	2	10	3
114	4	4	0	0	2	3	0	0	0	3	0.00	1	2	1
115	11	69	16	26	2	3	0	0	0	10	4037.20	1	2	1
116	15	72	18	51	15	12	0	1	4	27	13184.79	5	15	6
117	11	159	53	100	6	6	0	1	1	23	16126.42	2	6	3
118	9	52	17	22	2	3	0	0	0	10	2025.61	1	2	1
119	4	4	1	1	2	3	0	0	0	3	23.22	1	2	1
120	15	32	7	11	5	5	0	1	1	8	2259.98	2	5	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
121	15	26	5	8	5	5	0	1	1	6	1763.35	2	5	3
122	11	17	7	10	6	6	0	1	1	8	884.62	2	6	3
123	18	54	12	34	11	10	0	2	2	17	11011.06	3	11	5
124	9	18	11	12	2	3	0	0	0	10	636.50	1	2	1
125	18	35	8	14	9	8	1	2	2	9	3627.56	3	10	5
126	16	25	7	11	5	5	1	1	1	6	2047.23	2	6	3
127	8	10	4	6	5	5	2	1	1	6	344.16	2	7	3
รวม										1740	1249100.26	390	1882	653



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงผลการวัดค่าความซับซ้อนของระบบงานที่ 5

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
1	6	6	1	1	2	3	0	0	0	3	58.95	1	2	1
2	13	56	18	29	2	3	0	0	0	13	4409.92	1	2	1
3	8	8	3	3	2	3	0	0	0	3	152.21	1	2	1
4	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
5	38	168	30	70	42	33	22	10	11	49	64230.85	11	64	22
6	23	47	10	18	6	6	3	1	2	13	6787.23	2	9	4
7	10	67	26	37	26	16	0	1	1	28	3825.74	12	26	3
8	8	9	3	3	2	3	0	0	0	3	166.05	1	2	1
9	7	8	2	2	2	3	0	0	0	4	110.95	1	2	1
10	10	31	7	9	2	3	0	0	0	5	1051.06	1	2	1
11	12	35	8	10	2	3	0	0	0	4	1458.65	1	2	1
12	34	276	22	149	49	39	20	10	15	69	284171.03	12	69	26
13	27	83	21	35	20	17	6	4	4	33	14828.08	5	26	9
14	28	109	16	63	20	17	9	4	4	31	51763.60	5	29	9
15	10	15	4	4	2	3	0	0	0	3	361.70	1	2	1
16	10	21	7	9	2	3	2	0	0	4	788.30	1	4	1
17	30	88	27	51	19	16	7	4	6	30	22971.87	5	26	11
18	14	24	10	12	2	3	0	0	0	10	1386.49	1	2	1
19	6	7	2	2	2	3	0	0	0	3	81.00	1	2	1
20	8	15	4	4	2	3	0	0	0	5	272.46	1	2	1
21	7	8	0	0	2	3	0	0	0	4	0.00	1	2	1
22	19	69	16	28	24	17	1	3	5	22	8271.61	9	25	9
23	12	14	3	4	5	5	1	1	1	5	562.59	2	6	3
24	9	29	13	16	2	3	0	0	0	7	1111.43	1	2	1
25	8	8	3	3	2	3	0	0	0	3	152.21	1	2	1
26	12	14	3	4	5	5	1	1	1	5	562.59	2	6	3
27	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
28	17	45	8	15	10	9	0	2	2	14	4440.69	3	10	5
29	10	13	3	3	2	3	0	0	0	4	296.04	1	2	1
30	9	30	13	16	2	3	0	0	0	7	1136.13	1	2	1
31	9	11	4	4	2	3	0	0	0	4	249.78	1	2	1
32	12	14	3	4	5	5	1	1	1	5	562.59	2	6	3
33	12	14	3	4	5	5	1	1	1	5	562.59	2	6	3
34	7	10	1	2	2	3	0	0	0	3	252.00	1	2	1
35	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
36	23	69	14	24	21	18	5	4	5	32	9551.16	5	26	10
37	24	73	18	24	20	17	2	4	6	22	8368.88	5	22	11
38	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
39	18	46	13	17	11	10	2	2	2	14	3673.35	3	13	5
40	12	29	7	11	2	3	0	0	0	6	1602.08	1	2	1
41	12	15	4	5	6	6	0	1	1	6	600.00	2	6	3
42	10	15	3	3	2	3	0	0	0	3	333.04	1	2	1
43	9	13	3	3	2	3	0	0	0	3	258.12	1	2	1
44	8	24	11	13	2	3	0	0	0	6	743.00	1	2	1
45	8	8	3	3	2	3	0	0	0	3	152.21	1	2	1
46	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
47	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
48	12	14	3	4	5	5	1	1	1	5	562.59	2	6	3
49	8	11	1	3	2	3	0	0	0	4	532.55	1	2	1
50	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
51	6	6	1	1	2	3	0	0	0	3	58.95	1	2	1
52	6	6	1	1	2	3	0	0	0	3	58.95	1	2	1
53	6	6	1	1	2	3	0	0	0	3	58.95	1	2	1
54	6	6	1	1	2	3	0	0	0	3	58.95	1	2	1
55	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
56	7	9	3	3	2	3	0	0	0	4	139.52	1	2	1
57	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
58	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
59	26	77	21	41	15	13	5	3	3	22	16635.73	4	20	7
60	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
61	15	32	10	19	6	6	0	1	1	8	3374.92	2	6	3
62	6	6	1	1	2	3	0	0	0	4	58.95	1	2	1
63	6	7	1	1	2	3	0	0	0	3	67.38	1	2	1
64	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
65	6	7	1	1	2	3	0	0	0	3	67.38	1	2	1
66	15	32	10	19	6	6	0	1	1	8	3374.92	2	6	3
67	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
68	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
69	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
70	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
71	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
72	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
73	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
74	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
75	15	32	10	19	6	6	0	1	1	8	3374.92	2	6	3
76	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
77	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
78	8	9	4	4	2	3	0	0	0	5	186.42	1	2	1
79	12	16	4	4	6	6	0	1	1	7	480.00	2	6	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
80	6	7	1	1	2	3	0	0	0	3	67.38	1	2	1
81	12	16	4	4	6	6	0	1	1	7	480.00	2	6	3
82	6	7	1	1	2	3	0	0	0	3	67.38	1	2	1
83	6	6	1	1	2	3	0	0	0	3	58.95	1	2	1
84	6	7	1	1	2	3	0	0	0	3	67.38	1	2	1
85	6	7	1	1	2	3	0	0	0	3	67.38	1	2	1
86	13	36	13	21	2	3	0	0	0	14	2813.21	1	2	1
87	20	28	4	6	6	6	0	1	2	9	2338.33	2	6	4
88	6	6	1	1	2	3	0	0	0	3	58.95	1	2	1
89	6	7	1	1	2	3	0	0	0	3	67.38	1	2	1
90	58	288	48	134	91	58	6	9	9	104	229855.48	35	97	19
91	15	27	6	8	5	5	2	1	1	5	1537.31	2	7	3
92	15	27	6	8	5	5	2	1	1	5	1537.31	2	7	3
93	10	13	3	4	2	3	0	0	0	5	419.38	1	2	1
94	12	14	3	4	5	5	1	1	1	5	562.59	2	6	3
95	14	18	5	6	7	7	1	1	1	8	856.38	2	8	3
96	12	14	3	4	5	5	1	1	1	5	562.59	2	6	3
97	6	7	1	1	2	3	0	0	0	4	67.38	1	2	1
98	11	23	9	11	2	3	0	0	0	6	987.80	1	2	1
99	12	14	3	4	5	5	1	1	1	5	562.59	2	6	3
100	6	7	1	1	2	3	0	0	0	4	67.38	1	2	1
101	6	7	1	1	2	3	0	0	0	4	67.38	1	2	1
102	12	14	3	4	5	5	1	1	1	5	562.59	2	6	3
103	12	14	3	4	5	5	1	1	1	5	562.59	2	6	3
104	11	13	2	3	6	6	1	1	1	6	488.46	2	7	3
105	6	6	0	0	2	3	0	0	0	3	0.00	1	2	1
106	11	20	6	8	8	7	1	1	1	7	839.29	3	9	3
107	6	7	1	1	2	3	0	0	0	3	67.38	1	2	1
108	13	17	3	5	6	6	1	1	1	7	953.33	2	7	3
109	13	17	4	5	6	6	1	1	1	7	730.63	2	7	3
110	13	17	4	5	6	6	1	1	1	7	730.63	2	7	3
111	11	20	6	8	8	7	1	1	1	10	839.29	3	9	3
112	12	37	17	23	2	3	0	0	0	8	2366.12	1	2	1
113	16	22	6	6	6	6	0	1	1	6	998.91	2	6	3
114	11	16	4	4	2	3	0	0	0	4	429.76	1	2	1
115	28	108	14	28	23	19	0	5	8	24	20533.94	6	23	14
116	49	299	70	134	50	41	30	12	21	76	140017.89	11	80	34
117	18	33	5	13	7	7	1	1	2	10	4869.16	2	8	4
118	7	10	1	2	2	3	0	0	0	3	252.00	1	2	1
119	7	10	1	2	2	3	0	0	0	3	252.00	1	2	1
120	5	5	1	1	2	3	0	0	0	4	38.77	1	2	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
121	6	7	1	1	2	3	0	0	0	4	67.38	1	2	1
122	11	13	2	3	6	6	1	1	1	6	488.46	2	7	3
123	6	7	1	1	2	3	0	0	0	4	67.38	1	2	1
124	8	26	11	14	2	3	0	0	0	6	865.03	1	2	1
125	8	8	3	3	2	3	0	0	0	3	152.21	1	2	1
126	5	5	1	1	2	3	0	0	0	4	38.77	1	2	1
127	6	7	1	1	2	3	0	0	0	4	67.38	1	2	1
128	11	39	15	20	2	3	0	0	0	10	2033.72	1	2	1
129	6	8	4	4	2	3	0	0	0	4	119.59	1	2	1
130	13	24	11	15	7	7	2	1	1	11	1584.94	2	9	3
131	6	9	4	4	2	3	0	0	0	4	129.56	1	2	1
132	16	26	6	10	6	6	2	1	4	8	2140.53	2	8	6
133	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
134	12	28	10	14	2	3	0	0	0	12	1573.29	1	2	1
135	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
136	6	7	1	1	2	3	0	0	0	4	67.38	1	2	1
137	6	7	2	2	2	3	0	0	0	5	81.00	1	2	1
138	6	7	1	1	2	3	0	0	0	4	67.38	1	2	1
139	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
140	7	12	5	5	2	3	0	0	0	5	213.31	1	2	1
141	8	57	18	18	2	3	0	0	0	19	1410.13	1	2	1
142	9	11	2	2	2	3	0	0	0	4	202.38	1	2	1
143	5	5	2	2	2	3	0	0	0	3	49.13	1	2	1
144	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
145	6	7	1	1	2	3	0	0	0	4	67.38	1	2	1
146	6	6	3	3	2	3	0	0	0	4	85.59	1	2	1
147	10	12	3	5	5	6	1	1	1	8	524.23	1	6	3
148	15	26	6	11	7	7	3	1	1	13	2234.59	2	10	3
149	6	7	2	2	2	3	0	0	0	4	81.00	1	2	1
150	7	10	3	4	2	3	0	0	0	6	217.03	1	2	1
151	6	7	2	2	2	3	0	0	0	4	81.00	1	2	1
152	6	8	4	4	2	3	0	0	0	5	119.59	1	2	1
153	8	26	11	14	2	3	0	0	0	6	865.03	1	2	1
154	8	8	3	3	2	3	0	0	0	3	152.21	1	2	1
155	15	26	6	11	7	7	3	1	1	13	2234.59	2	10	3
156	6	6	3	3	2	3	0	0	0	4	85.59	1	2	1
157	6	7	1	1	2	3	0	0	0	4	67.38	1	2	1
158	6	7	1	1	2	3	0	0	0	4	67.38	1	2	1
159	6	7	1	1	2	3	0	0	0	4	67.38	1	2	1
160	6	6	3	3	2	3	0	0	0	4	85.59	1	2	1
161	10	13	3	5	5	6	1	1	1	8	555.07	1	6	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
162	13	30	10	15	6	6	5	1	1	14	1984.71	2	11	3
163	10	13	3	5	5	6	1	1	1	8	555.07	1	6	3
164	11	26	6	9	2	3	0	0	0	9	1180.25	1	2	1
165	6	8	4	4	2	3	0	0	0	5	119.59	1	2	1
166	8	26	11	14	2	3	0	0	0	6	865.03	1	2	1
167	8	8	2	2	2	3	0	0	0	3	132.88	1	2	1
168	6	7	1	1	2	3	0	0	0	4	67.38	1	2	1
169	6	6	2	2	2	3	0	0	0	3	72.00	1	2	1
170	8	8	3	3	2	3	0	0	0	3	152.21	1	2	1
171	12	37	14	22	16	11	1	1	1	18	2614.79	7	17	3
172	7	10	1	2	2	3	0	0	0	3	252.00	1	2	1
173	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
174	15	21	5	7	5	5	1	1	2	7	1270.65	2	6	4
175	7	10	1	2	2	3	0	0	0	3	252.00	1	2	1
176	7	10	3	3	2	3	0	0	2	4	151.15	1	2	3
177	8	8	3	3	2	3	0	0	2	3	152.21	1	2	3
178	5	5	1	1	2	3	0	0	2	3	38.77	1	2	3
179	8	10	2	2	2	3	0	0	2	4	159.45	1	2	3
180	9	12	3	4	2	3	0	0	2	4	344.16	1	2	3
181	9	12	3	4	2	3	0	0	2	4	344.16	1	2	3
182	9	12	3	4	2	3	0	0	2	4	344.16	1	2	3
183	9	12	3	4	2	3	0	0	2	4	344.16	1	2	3
184	9	12	3	4	2	3	0	0	2	4	344.16	1	2	3
รวม										1421	981183.04	344	1105	468

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงผลการวัดค่าความซับซ้อนของระบบงานที่ 6

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
1	29	52	16	24	17	14	3	4	7	18	9078.03	5	20	12
2	7	11	4	4	2	3	0	0	0	4	181.62	1	2	1
3	7	11	4	4	2	3	0	0	0	4	181.62	1	2	1
4	34	61	17	27	14	12	5	3	3	20	13477.68	4	19	7
5	28	50	16	26	13	11	0	3	3	27	9439.36	4	13	7
6	16	25	8	10	7	7	0	1	4	10	1604.74	2	7	6
7	35	113	22	55	39	31	11	7	10	52	42871.74	10	50	18
8	8	10	3	3	2	3	0	0	0	5	179.89	1	2	1
9	11	15	6	6	2	3	0	0	0	7	472.10	1	2	1
10	27	66	23	33	9	9	0	2	2	26	10822.59	2	9	5
11	23	74	23	35	20	16	1	3	5	31	10536.19	6	21	9
12	21	50	15	19	19	15	0	2	4	23	4744.44	6	19	7
13	16	20	10	12	10	9	0	2	2	14	1443.98	3	10	5
14	12	17	6	7	7	7	0	1	1	9	700.55	2	7	3
15	12	15	7	8	7	7	2	1	1	10	669.96	2	9	3
16	15	22	6	6	12	10	0	2	2	12	922.39	4	12	5
17	17	21	5	5	6	6	0	1	1	9	985.53	2	6	3
18	9	17	8	9	2	3	0	0	0	8	538.01	1	2	1
19	11	15	6	6	2	3	0	0	0	7	472.10	1	2	1
20	15	37	17	21	2	3	0	0	0	11	2686.76	1	2	1
21	19	30	8	13	11	10	1	2	2	17	3156.35	3	12	5
22	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
23	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
24	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
25	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
26	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
27	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
28	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
29	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
30	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
31	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
32	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
33	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
34	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
35	27	68	16	17	30	18	1	1	2	32	6615.80	14	31	4
36	29	72	23	30	26	21	0	4	6	33	10996.89	7	26	11
37	9	10	4	4	2	3	0	0	0	6	233.13	1	2	1
38	9	10	4	4	2	3	0	0	0	6	233.13	1	2	1
39	20	31	7	13	11	10	1	2	2	17	3885.42	3	12	5
รวม										457	128051.98	101	327	137

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงผลการวัดค่าความซับซ้อนของระบบงานที่ 7

No.	n1	N1	n2	N2	edge	node	df	CFS	CFD	LOC	Halstead	McCabe	Oviedo	SDM
1	17	31	9	14	10	9	1	2	4	12	2796.76	3	11	7
2	17	34	9	17	12	10	1	2	2	12	3848.88	4	13	5
3	13	25	7	9	6	6	2	1	3	8	1228.04	2	8	5
4	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
5	6	7	1	1	2	3	0	0	0	3	67.38	1	2	1
6	8	10	3	4	2	3	0	0	0	6	258.30	1	2	1
7	8	10	3	4	2	3	0	0	0	6	258.30	1	2	1
8	16	49	12	19	10	9	0	2	5	13	4140.74	3	10	8
9	7	22	12	14	2	3	0	0	0	10	624.45	1	2	1
10	6	12	8	8	2	3	0	0	0	6	228.44	1	2	1
11	18	37	12	13	7	7	0	1	1	12	2392.11	2	7	3
12	7	8	1	1	2	3	0	0	0	4	94.50	1	2	1
13	8	10	3	4	2	3	0	0	0	6	258.30	1	2	1
14	7	39	15	23	2	3	0	0	0	14	1483.80	1	2	1
15	9	48	18	26	2	3	0	0	0	16	2287.10	1	2	1
16	8	8	3	3	2	3	0	0	0	5	152.21	1	2	1
17	7	36	14	21	2	3	0	0	0	13	1314.40	1	2	1
18	8	10	3	4	2	3	0	0	0	6	258.30	1	2	1
19	7	8	2	2	2	3	0	0	0	3	110.95	1	2	1
20	8	12	3	3	2	3	0	0	0	4	207.57	1	2	1
21	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
22	7	8	2	2	2	3	0	0	0	3	110.95	1	2	1
23	5	5	0	0	2	3	0	0	0	3	0.00	1	2	1
24	7	8	2	2	2	3	0	0	0	3	110.95	1	2	1
25	16	49	12	19	10	9	0	2	5	13	4140.74	3	10	8
26	7	8	3	3	2	3	0	0	0	4	127.89	1	2	1
27	7	8	3	3	2	3	0	0	0	4	127.89	1	2	1
28	7	8	3	3	2	3	0	0	0	4	127.89	1	2	1
29	9	35	16	23	2	3	0	0	0	9	1742.32	1	2	1
30	8	12	4	6	2	3	0	0	0	5	387.18	1	2	1
31	5	5	1	1	2	3	0	0	0	3	38.77	1	2	1
32	13	40	14	25	24	19	18	6	9	20	3587.39	7	42	16
33	18	90	31	62	29	23	21	7	10	34	15361.85	8	50	18
34	21	55	11	15	6	6	0	1	3	9	5011.36	2	6	5
35	31	253	32	119	42	33	19	7	17	68	128166.70	11	61	25
36	9	31	7	10	2	3	0	0	0	9	1054.29	1	2	1
37	10	14	4	4	5	5	0	1	1	6	342.66	2	5	3
38	6	27	14	18	2	3	0	0	0	11	750.16	1	2	1
39	19	53	19	27	6	6	1	1	3	18	5667.76	2	7	5
รวม										391	188867.30	76	284	135

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.

หน้าจอแสดงผลความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM

1. หน้าจอแสดงผลการวัดความซับซ้อนจากตัววัดของ Halstead ของ โปรแกรมรูปที่ 3.2 (ก) และ 3.2 (ข)

Sub Name : Sub Text1_Change()

Operator	Frequency
1 Sub	1
2 ()	2
3 "	8
4 Val	1
5 If	1
6 Then	7
7 Print	8
8 ""	8
9 Elseif	6
10 Else	1
11 End If	1
12 End Sub	1

Total foundkey (N1) : 45

Operand	Frequency
1 1	8
2 Text1	1
3 1	1
4 "รับค่า"	1
5 2	1
6 "รับค่า"	1
7 3	1
8 "รับค่า"	1
9 4	1
10 "รับค่า"	1
11 5	1
12 "รับค่า"	1
13 6	1
14 "รับค่า"	1
15 7	1
16 "รับค่า"	1
17 "เลขไม่ถูกต้อง"	1

Total foundkey (N2) : 24

Sub Name : Sub Text2_Change()

Operator	Frequency
1 Sub	1
2 ()	2
3 "	1
4 Val	1
5 Select Case	1
6 Case	8
7 Print	8
8 ""	8
9 Else	1
10 End Select	1
11 End Sub	1

Total foundkey (N1) : 33

Operand	Frequency
1 1	2
2 Text1	1
3 1	1
4 "รับค่า"	1
5 2	1
6 "รับค่า"	1
7 3	1
8 "รับค่า"	1
9 4	1
10 "รับค่า"	1
11 5	1
12 "รับค่า"	1
13 6	1
14 "รับค่า"	1
15 7	1
16 "รับค่า"	1
17 "เลขไม่ถูกต้อง"	1

Total foundkey (N2) : 18

Auto

Filename : d:\Progub\test1.frm
Total Sub Number : 2

Sub Name	Halstead
Sub Text1_Change()	2839.347
Sub Text2_Change()	1427.784

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

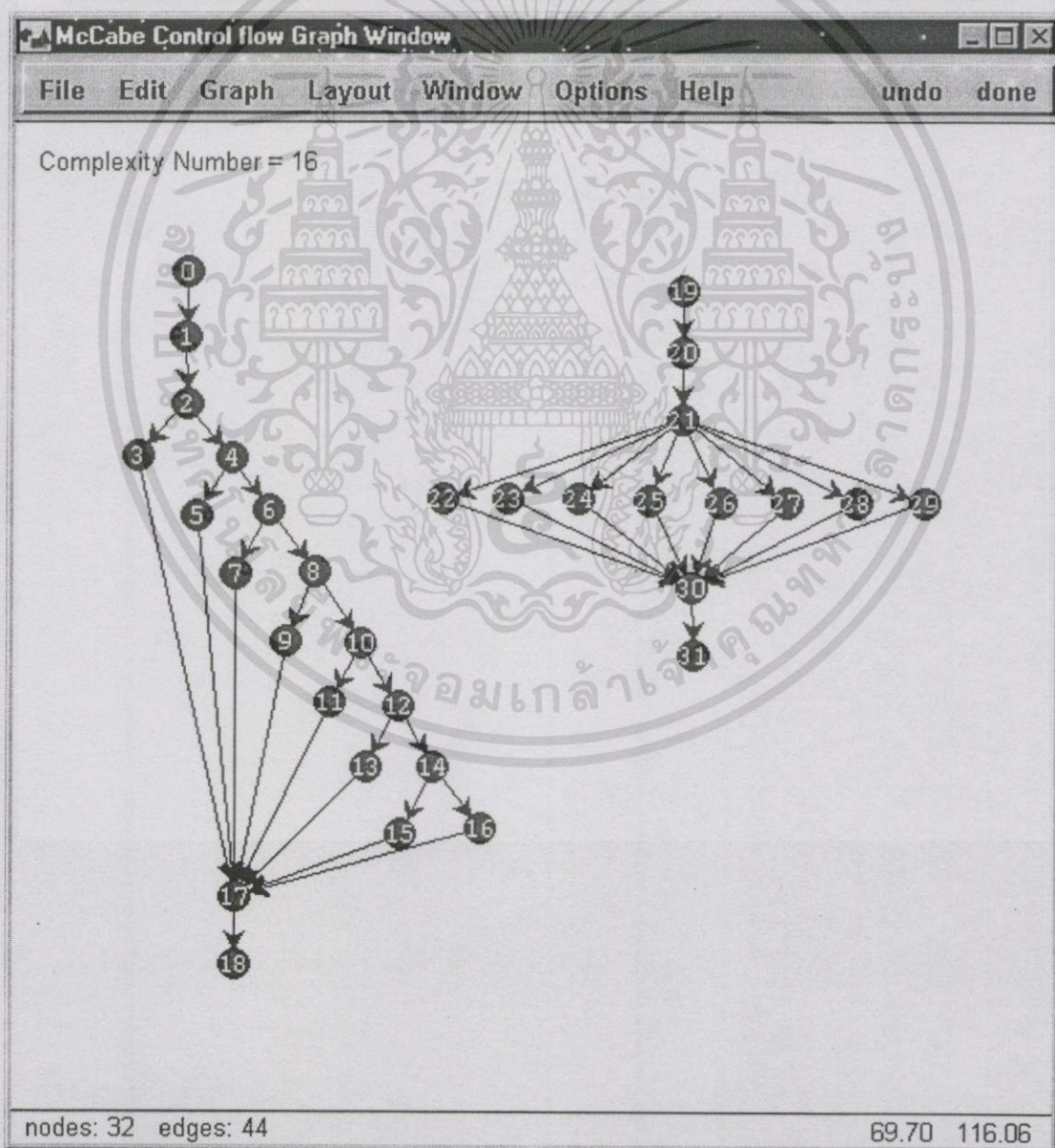
2. หน้าจอแสดงผลการวัดความซับซ้อนจากตัววัดของ McCabe ของโปรแกรมรูปที่ 3.2 (ก) และ 3.2 (ข)

McCabe

Auto

Filename : d:\Progub\test1.frm
Total Sub Number : 2

Sub Name	Edge	Node	LOC	McCabe
Sub Text1_Change()	25	19	20	8
Sub Text2_Change()	19	13	21	8
Total			41	16



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. หน้าจอแสดงผลการวัดความซับซ้อนจากตัววัดของ Oviedo ของโปรแกรมรูปที่ 3.2 (ก) และ 3.2 (ข)

Oviedo

Auto

Sub Name : Sub Text1_Change()

Node	ExpVar	VarDef
2	1	1(1)
4	1	1(1)
6	1	1(1)
8	1	1(1)
10	1	1(1)
12	1	1(1)
14	1	1(1)

ExpTotal = 7

Oviedo

Auto

Sub Name : Sub Text2_Change()

Node	ExpVar	VarDef
21	1	1(28)

ExpTotal = 1

Total Sub Number : 2

Sub Name	ExpVar	Edge	Nodes
Sub Text1_Change()	7	25	32
Sub Text2_Change()	1	19	28
Total			52

4. หน้าจอแสดงผลการวัดความซับซ้อนจากตัววัด SDM ของโปรแกรมรูปที่ 3.2 (ก) และ 3.2 (ข)

SDM

Auto

Enter File name : d:\Progub\test1.frn

1 Sub Name -> Sub Text1_Change()

node	Control Flow Data
2	I
4	I
6	I
8	I
10	I
12	I
14	I

CFD = 7

CFS = 1

SDM = 9

SDM

Auto

2 Sub Name -> Sub Text2_Change()

node	Control Flow Data
21	I

CFD = 1

CFS = 1

SDM = 3

Filename : d:\Progub\test1.frn

Total Sub Number : 2

Sub Name	CFS	CFD	SDM
Sub Text1_Change()	1	7	9
Sub Text2_Change()	1	1	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ.

บทความวิชาการและผลงานที่ได้รับการตีพิมพ์

1. นิภาพร ประภาศิริ และเอื้อน ปิ่นเงิน. "การวัดความซับซ้อนของซอฟต์แวร์." สารสนเทศลาดกระบัง, ปีที่ 3, ฉบับที่ 1, 2541.
2. นิภาพร ประภาศิริ และเอื้อน ปิ่นเงิน. "พื้นฐานทฤษฎีการวัดซอฟต์แวร์." วารสาร E@SI, ฉบับที่ 8, 2544.
3. นิภาพร ประภาศิริ และเอื้อน ปิ่นเงิน. "ตัววัดความซับซ้อนของซอฟต์แวร์จากโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล." วิศวกรรมลาดกระบัง, ปีที่ 18, ฉบับที่ 2, 2544.





วารสาร

มกราคม 2541 ปีที่ 3 ฉบับที่ 1

สารสนเทศลาดกระบัง

Ladkrabang Information Journal

ISSN 0859 - 5208

January 1998 Vol. 3 No. 1

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
Faculty of Information Technology, King Mongkut's Institute of Technology, Ladkrabang, Bangkok 10520

- 1 การขยายการใช้ระบบฐานข้อมูลแบบสัมพันธ์เพื่อรองรับคำถามที่คลุมเครือ
Extensions of Relational Database Systems to Support Fuzzy Query
ดร. เอื้อน ปิ่นเงิน, พ.อ. ภิรมย์ ทิพย์สิงห์
- 14 การประมาณค่าความเอียงหน้าเอกสารสิ่งพิมพ์
The Skew Estimation of Printed Documents
ดร. บุทธพงษ์ รุ่งสรศักดิ์, กฤษณะ ชินสาร
- 22 การพัฒนาโปรแกรมคำนวณหาทิศทางยิงของปืนใหญ่ขนาดเบากระสุนวิถีโค้งด้วยระบบ PVM
Development of Fire-Directions Calculations Program for Light
Howitzer using Parallel Virtual Machine (PVM)
บรรจง ปิยะธำรง, เทพจิต อินทร์คำ
ไอฟาร์ วงศ์วิรัตน์
- 31 ระบบการเรียนการสอนทางไกลแบบสื่อประสมผ่านดาวเทียม
Multimedia Distance Learning System via Satellite
ไอฟาร์ วงศ์วิรัตน์
- 42 การวัดความซับซ้อนของซอฟต์แวร์
Software Complexity Measurement
น.ศ. นิภาพร ประภาศิริ, ดร. เอื้อน ปิ่นเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวัดความซับซ้อนของซอฟต์แวร์ (Software Complexity Measurement)

น.ส. นิภาพร ประภาศิริ*

ดร.เอื้อน ปิ่นเงิน**

บทคัดย่อ

การวัดคุณสมบัติของซอฟต์แวร์สามารถนำไปใช้ในการประเมินคุณภาพของซอฟต์แวร์ที่สร้างขึ้นมาได้ ช่วยให้เข้าใจสิ่งที่เกิดขึ้นระหว่างการพัฒนาและการบำรุงรักษาซอฟต์แวร์นั้น ดังนั้นจึงได้พยายามวัดคุณสมบัติต่าง ๆ ของซอฟต์แวร์ให้ออกมาเป็นตัวเลข แต่ก็มีคุณสมบัติบางอย่างที่ไม่สามารถวัดค่าออกมาได้โดยตรง จึงจำเป็นต้องหาความสัมพันธ์ระหว่างสิ่งที่เราสามารถวัดได้กับสิ่งที่เราต้องการทราบ นักวิจัยหลายท่านพยายามค้นหาความสัมพันธ์ระหว่างลักษณะของโปรแกรมกับความยากในการสร้างโปรแกรม การวัดความซับซ้อนของซอฟต์แวร์สามารถใช้ในการประเมินต้นทุนของโครงการ การจัดสรรกำลังคนในการพัฒนารวมถึงการประเมินค่าใช้จ่ายในการบำรุงรักษาอีกด้วย บทความนี้ได้นำเสนอถึงวิธีการวัดความซับซ้อนของซอฟต์แวร์แบบต่างๆ โดยพิจารณาถึงคุณสมบัติที่สำคัญของโปรแกรมคือ การวัดความซับซ้อนจากโครงสร้าง (structure complexity measurement) และการวัดความซับซ้อนจากขนาดของโปรแกรม (size) การวัดความซับซ้อนจากโครงสร้างแบ่งออกเป็น 3 ส่วนคือ โครงสร้างควบคุมการไหล (control-flow structure) โครงสร้างการไหลของข้อมูล (data flow structure) และโครงสร้างข้อมูล (data structure) คุณสมบัติเหล่านี้อาจมีผลต่อคุณสมบัติภายนอกของซอฟต์แวร์ เช่น การบำรุงรักษา (maintainability) การทดสอบ (testability) การนำกลับมาใช้ใหม่ (reusability) และความน่าเชื่อถือ (reliability) เป็นต้น

Abstract

Software measurement can be used to evaluate the quality of the software project. It helps us understand what could be happening during development and maintenance of software product. The attempt measured of the attributes are numerical values but some attributes may not be easily quantified. We should find the relationship between what it can be measured and what we want to know. Researchers have attempted to find relationship between the characteristics of programs and the difficulty of performing programming tasks. Measurement of software complexity can be used for cost projection, manpower allocation, and maintenance. This paper presents the methods for software complexity measurement by considering the main attributes of programs such as size complexity measurement and structure complexity measurement. Structure complexity measurement can be classified into 3 parts: control flow structure, data flow structure, and data structure. These attributes can be affected to external attributes of software such as maintainability, testability, reusability and reliability.

* นักศึกษาปริญญาโท คณะเทคโนโลยีสารสนเทศ

** อาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬ.

1. บทนำ

การประเมินคุณภาพของซอฟต์แวร์ จะต้องพิจารณาคุณลักษณะต่าง ๆ ของซอฟต์แวร์ประกอบกัน เช่น การบำรุงรักษา การทดสอบ การนำกลับมาใช้ใหม่ และ ความน่าเชื่อถือ เป็นต้น คุณสมบัติเหล่านี้มีผลกระทบต่อวัฏจักรการพัฒนาซอฟต์แวร์ การวัดคุณสมบัติเหล่านี้โดยตรงเป็นสิ่งที่ยาก ต้องพิจารณาถึงคุณสมบัติภายในของ ซอฟต์แวร์แทน แล้วสร้างความสัมพันธ์ของคุณสมบัติภายในที่สามารถวัดได้กับคุณสมบัติภายนอก โดยศึกษาจากหลักการและกฎเกณฑ์ต่าง ๆ ที่มีอยู่ นักวิศวกรรมซอฟต์แวร์หลายท่านกล่าวว่า คุณสมบัติภายในที่แสดงถึงความซับซ้อนของซอฟต์แวร์ สามารถนำไปใช้พยากรณ์คุณสมบัติภายนอกต่าง ๆ ได้ถูกต้อง [5]

บทความนี้นำเสนอวิธีการวัดความซับซ้อนของซอฟต์แวร์แบบต่าง ๆ โดยพิจารณาถึงคุณสมบัติภายในของโปรแกรมเป็นหลักคือ การวัดความซับซ้อนจากโครงสร้าง และการวัดความซับซ้อนจากขนาดของโปรแกรม การวัดความซับซ้อนจากโครงสร้างแบ่งออกเป็น โครงสร้างควบคุมการไหล โครงสร้างการไหลของข้อมูล และ โครงสร้างข้อมูล พร้อมทั้งนำเสนอตัวอย่างการวัดความซับซ้อนด้วยวิธีต่าง ๆ เมื่อสามารถวัดค่าความซับซ้อนของซอฟต์แวร์ได้ก็จะสามารถประเมินต้นทุนโครงการได้ สามารถจัดสรรกำลังคนในการพัฒนา รวมถึงการประเมินค่าใช้จ่ายในการบำรุงรักษาได้ด้วย

2. การวัดความซับซ้อนจากโครงสร้าง

การวัดความซับซ้อนจากโครงสร้างแบ่งเป็น 3 ส่วน คือ โครงสร้างควบคุมการไหล โครงสร้างการไหลของข้อมูล และ โครงสร้างข้อมูล ซึ่งมีรายละเอียดดังนี้

2.1 โครงสร้างควบคุมการไหล

โครงสร้างควบคุมการไหลคือ การเตรียมลำดับของคำสั่งในการทำงานของโปรแกรม รูปแบบของโครงสร้างแสดงให้เห็นซ้ำ ๆ กัน ทำให้การทำงานของโปรแกรมเป็นแบบวนรูป ซึ่งในการนับจำนวนของคำสั่งจะนับเพียงครั้งเดียวเท่านั้น แต่ในการควบคุมการไหล เราจะกระทำหลายครั้งให้เหมือนกับโปรแกรมกำลังทำงานอยู่จริง เพราะคำสั่งหนึ่งอาจมีการทำงานหลายครั้ง

การวัดคุณสมบัติซอฟต์แวร์แบบต่าง ๆ ให้ความสนใจในการวัดโครงสร้างควบคุมการไหลจากภาษาที่ใช้ในการเขียนโปรแกรมหรือ อัลกอริทึม การวัดการควบคุมการไหลเป็นการจำลองด้วยกราฟ ที่มีทิศทาง โดย โหนดหนึ่งตรงกับหนึ่งคำสั่งในโปรแกรม และแต่ละ arc จะแสดงการไหลของการควบคุมจากโหนดหนึ่งไปยังโหนดถัดไป เรียกกราฟดังกล่าวว่า กราฟควบคุมการไหล (control flow graphs) หรือ โฟลว์กราฟ (flowgraphs) จากโปรแกรมรูปที่ 1(a) แสดงโฟลว์กราฟของโปรแกรมได้ดังรูปที่ 1 (b)

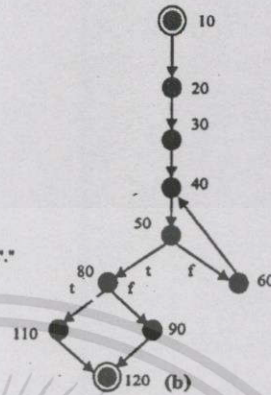
โฟลว์กราฟ คือ กราฟที่มีทิศทางจากโหนดเริ่มต้น (start node) ถึงโหนดสิ้นสุด (stop node) โดยที่โหนดสิ้นสุดจะไม่มีทิศทางออกจากตัวมันเอง โฟลว์กราฟเป็นโมเดลที่เหมาะสมสำหรับการวัดโครงสร้างการควบคุมการไหลของโปรแกรม เพราะแสดงให้เห็นโครงสร้างของโปรแกรมได้อย่างชัดเจน โดย โหนดแสดงถึงคำสั่งในโปรแกรม และ arc แสดงรูปแบบการควบคุมการไหลได้อย่างชัดเจน

ถ้าให้ m แทนการวัดโครงสร้างที่กำหนดในโฟลว์กราฟ และถ้า A เป็นโปรแกรมที่มีโครงสร้างที่ซับซ้อนกว่าโปรแกรม B แล้ว $m(A)$ จะต้องมากกว่า $m(B)$ ในการวิเคราะห์โครงสร้างควบคุมการไหล วิธีนี้สามารถแสดงให้เห็นได้ว่าโปรแกรมที่เป็น

สารสนเทศลาดกระบัง

```

10 INPUT P
20 DIV = 2
30 Lim = INT(SQR(P))
40 Flag = P/DIV - INT(P/DIV)
50 IF Flag = 0 OR DIV = Lim THEN 80
60 DIV = DIV + 1
70 GO TO 40
80 IF Flag <> 0 OR P > 4 THEN 110
90 PRINT DIV; "Smallest factor of"; P; "."
100 GO TO 120
110 PRINT P; " is prime"
120 END
    
```

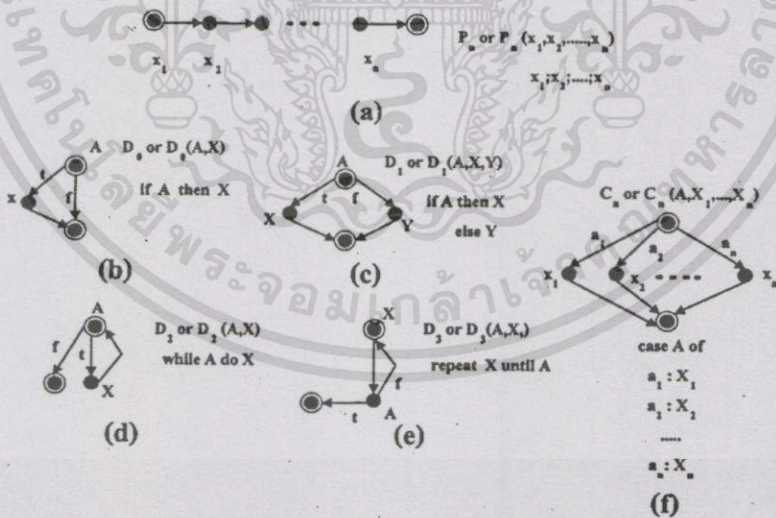


รูปที่ 1 (a) โปรแกรม (b) โฟลว์กราฟของโปรแกรม

โครงสร้างสามารถแยกออกเป็น ส่วน ๆ อย่างชัดเจน และจะมีเพียงรูปแบบเดียวเท่านั้น

จากคุณสมบัติ การควบคุมการไหลของโปรแกรม จะเห็นว่าโปรแกรมหนึ่ง ๆ สามารถแทนด้วยโปรแกรมโฟลว์กราฟได้เพียงรูปแบบเดียว โดยโฟลว์กราฟดังกล่าวประกอบด้วยโครงสร้างเฉพาะ (prime structures) ซึ่งเป็นโครงสร้างที่สนับสนุน

การพัฒนาโปรแกรมที่เป็นโปรแกรมโครงสร้าง โครงสร้างเฉพาะประกอบด้วยโฟลว์กราฟเฉพาะ (prime flowgraphs) หลาย ๆ โฟลว์กราฟประกอบกัน โฟลว์กราฟเฉพาะ คือ โฟลว์กราฟที่ไม่สามารถแตกย่อยไปเป็นการทำงานแบบลำดับ (sequence) หรือการรวมกลุ่ม (nesting) ได้อีก เช่น โฟลว์กราฟในรูปที่ 2 แสดงโครงสร้างของโฟลว์กราฟเฉพาะแบบต่าง ๆ



รูปที่ 2 (a) โฟลว์กราฟ P_n (b) โฟลว์กราฟ D_0
(c) โฟลว์กราฟ D_1 (d) โฟลว์กราฟ D_2
(e) โฟลว์กราฟ D_3 (f) โฟลว์กราฟ C_n

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพลีกราฟรูปที่ 2 (a) เรียกว่า P_n เป็นโพลีกราฟแสดงการไหลของโปรแกรมที่มีการทำงานแบบลำดับจากโนดที่ x_1 ถึงโนดที่ x_n โพลีกราฟรูปที่ 2 (b) ถึงรูปที่ 2 (e) แสดงโครงสร้างของโพลีกราฟเฉพาะที่เรียกว่าโครงสร้าง D ซึ่งเป็นโครงสร้างของโพลีกราฟที่มีเงื่อนไขของโปรแกรมดังนี้

โพลีกราฟรูปที่ 2(b) เงื่อนไขของโปรแกรมเป็น if A then X เรียกว่าโพลีกราฟ D_0

โพลีกราฟรูปที่ 2 (c) เงื่อนไขของโปรแกรมเป็น if A then X else Y เรียกว่าโพลีกราฟ D_1

โพลีกราฟรูปที่ 2 (d) เงื่อนไขของโปรแกรมเป็น while A do X เรียกว่าโพลีกราฟ D_2

โพลีกราฟรูปที่ 2(e) เงื่อนไขของโปรแกรมเป็น repeat X until A เรียกว่าโพลีกราฟ D_3

ส่วนรูปที่ 2(ก) แสดงโพลีกราฟที่มีเงื่อนไขเป็น Case เรียกโพลีกราฟนี้ว่า C_n ซึ่งแสดง case A จาก $a_1 : x_1$ ถึง $a_n : x_n$

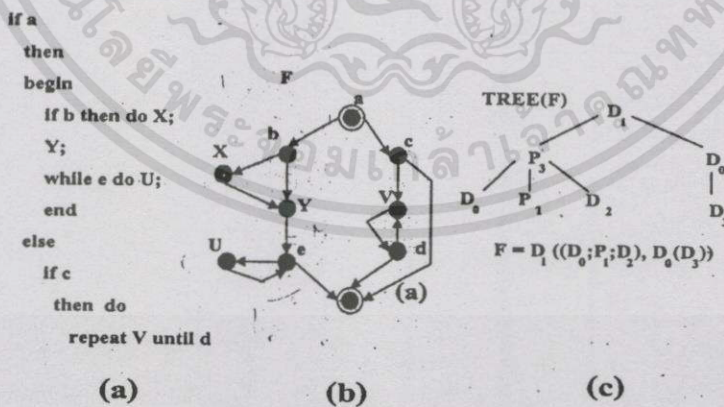
จากโพลีกราฟของโปรแกรม สามารถแปลงให้อยู่ในลักษณะของโครงสร้างต้นไม้เฉพาะ (prime decomposition tree) ได้ เช่น โปรแกรมในรูปที่ 3(a) สามารถเขียนเป็นโพลีกราฟได้ดังรูปที่ 3(b) แล้ว

แปลงให้อยู่ในรูปแบบของโครงสร้างต้นไม้เฉพาะได้ดังรูปที่ 3(c) การแปลงโปรแกรมให้เป็นโครงสร้างต้นไม้เฉพาะสามารถนำไปใช้ในการอธิบายว่าโพลีกราฟนั้นเกิดจากการทำงานแบบลำดับ (sequence) และการรวมกลุ่ม (nesting) อย่างไร และยังสามารถนำไปใช้ในการวัดลำดับชั้นของโพลีกราฟนั้นได้

สมมติว่าต้องการวัดความลึกของการรวมกลุ่มภายในโปรแกรมใดโปรแกรมหนึ่ง จะต้องเขียนโปรแกรมเป็นโพลีกราฟ (F) ก่อน แล้วจึงแตกออกเป็นโครงสร้างต้นไม้เฉพาะ และให้ α แทนความลึกของการรวมกลุ่ม เราสามารถกำหนดค่า α ในรูปแบบของ primes, sequence และ nesting ได้ดังนี้

Primes : ความลึกจากการรวมกลุ่มของ prime P_i เป็นศูนย์ และความลึกจากการรวมกลุ่มของ prime F อื่น ๆ เท่ากับ 1 ดังนั้นจะได้ $\alpha(P_i) = 0$ และถ้า F เป็น prime ที่ไม่เท่ากับ P_i แล้ว $\alpha(F) = 1$

Sequence : ความลึกของการรวมกลุ่มแบบลำดับ F_1, \dots, F_n เท่ากับค่าความลึกที่มากที่สุดจากการรวมกลุ่มของ F_i นั่นคือ $\alpha(F_1; \dots; F_n) = \max(\alpha(F_1), \dots, \alpha(F_n))$



รูปที่ 3 (a) ตัวอย่างโปรแกรม (b) โพลีกราฟของโปรแกรม (a) (c) โครงสร้างต้นไม้เฉพาะ (prime decomposition tree) ของโพลีกราฟ (b)

Nesting : ความลึกจากการรวมกลุ่มของโพลีกราฟ $F(F_1, \dots, F_n)$ เท่ากับ ค่าความลึกที่มากที่สุดจากการรวมกลุ่มของ F_i บวก 1 เนื่องจากระดับการรวมกลุ่มเพิ่มขึ้นอีก 1 ระดับใน F นั่นคือ

$$\alpha(F(F_1, \dots, F_n)) = 1 + \max(\alpha(F_1), \dots, \alpha(F_n))$$

พิจารณาโครงสร้างต้นไม้เฉพาะในรูปที่ 3 (c) เราทราบว่า $F = D_1((D_0; P_1; D_2), D_0(D_3))$ ดังนั้นสามารถคำนวณ $\alpha(F)$ ได้ดังนี้

$$\begin{aligned} \alpha(F) &= \alpha(D_1((D_0; P_1; D_2), D_0(D_3))) \\ &= 1 + \max(\alpha(D_0; P_1; D_2), \alpha(D_0(D_3))) \\ &= 1 + \max(\max(\alpha(D_0), \alpha(P_1), \alpha(D_2)), 1 + \alpha(D_3)) \\ &= 1 + \max(\max(1, 0, 1), 2) \\ &= 1 + \max(1, 2) = 3 \end{aligned}$$

จากโครงสร้างต้นไม้เฉพาะแสดงให้เห็นว่ามีการกำหนดต้นไม้เป็นระดับชั้นที่ใหญ่ที่สุดซึ่งเป็นสิ่งที่วัดทั้งหมด เราสามารถกำหนดการวัดได้หลายแบบ โดยการกำหนดค่าให้แก่ primes การทำงานแบบลำดับ และการรวมกลุ่ม

2.1.1 การวัดความซับซ้อนจากค่าไซโคลเมติกของ McCabe (McCabe's cyclomatic complexity measure)

MCCabe ได้นำเสนอการวัดความซับซ้อนของโปรแกรมโดยใช้ค่าไซโคลเมติก (cyclomatic number) วัดจากโพลีกราฟของโปรแกรม [4] จากโปรแกรมหนึ่งสามารถสร้างโพลีกราฟ F และคำนวณค่าไซโคลเมติกได้ดังนี้

$$v(F) = e - n + 2$$

โดยที่ F มี arcs จำนวน e เส้น และมี n โหนด

ค่าไซโคลเมติกเป็นการวัดจำนวนของเส้นทางที่เป็นอิสระเชิงเส้น (linearly independent paths) ของ F การวัดนี้จะเป็นไปตามวัตถุประสงค์และเป็นประโยชน์เมื่อนับเส้นทางที่เป็นอิสระเชิงเส้นได้ถูกต้อง ค่าไซโคลเมติกสามารถคำนวณได้อีกวิธีหนึ่งดังนี้

$$v(F) = 1 + d$$

โดยที่ d คือจำนวนโหนดที่มีทิศทางออกจากตัวเอง (out degree) มากกว่า 1 ซึ่งอยู่ใน F นั้น

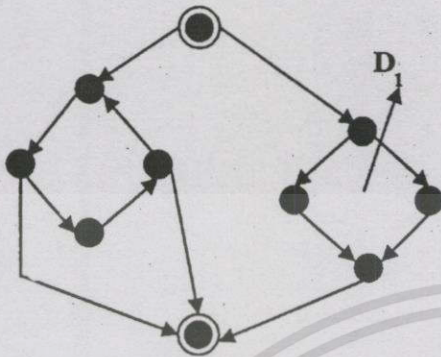
ทฤษฎีการวัดที่มีอยู่เกิดจากสมมติฐานที่ตั้งขึ้นจากความรู้สึกที่สัมพันธ์กับความซับซ้อน อย่างไรก็ตามได้มีการนำค่าไซโคลเมติกมาใช้แสดงให้เห็นถึงความยากของโปรแกรม McCabe แนะนำไว้ว่าค่า v ของโมดูลไม่ควรมีค่าเกิน 10 และโดยทั่วไปแล้วค่า v ที่เหมาะสมจะมีค่าอยู่ระหว่าง 3 ถึง 7

2.1.2 การวัดปัจจัยความซับซ้อนของ McCabe (McCabe's essential complexity measure)

MCCabe ได้เสนอการวัดความซับซ้อนของโปรแกรมอีกลักษณะหนึ่ง โดยพิจารณาจากระดับโครงสร้างของโปรแกรมทั้งหมดจากโพลีกราฟ และกำหนดค่าปัจจัยความซับซ้อน (ev) ไว้ดังนี้

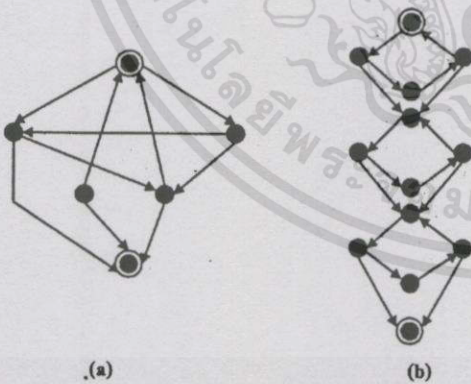
$$ev(F) = v(F) - m$$

โดยที่ m คือจำนวนของโพลีกราฟย่อยของ F ซึ่งเป็น D-structured prime หรือเรียกว่าโครงสร้าง D [4] โพลีกราฟ F ในรูปที่ 4 มีโพลีกราฟย่อยที่เป็นโครงสร้าง D เพียงโครงสร้างเดียวคือโครงสร้างของ D_1 ซึ่งอยู่ทางขวาของโพลีกราฟ ดังนั้นได้ $m = 1$ และ $v(F) = 5$ ทำให้ผลของปัจจัยความซับซ้อน $ev(F) = 5 - 1 = 4$



รูปที่ 4 โฟลว์กราฟแสดงค่าปัจจัยความซับซ้อนของโปรแกรมซึ่งมีค่าเท่ากับ 4

McCabe ชื่นชมว่าปัจจัยความซับซ้อนสามารถลดลงได้ ถ้าโฟลว์กราฟมีส่วนประกอบของโครงสร้าง D อยู่ จึงได้ลดค่าความซับซ้อนของโฟลว์กราฟที่มีโครงสร้าง D ประกอบอยู่ด้วย เมื่อโฟลว์กราฟไม่เป็นโครงสร้าง D นิยามการวัดปัจจัยความซับซ้อนจะไม่สอดคล้องกับโครงสร้างที่ซับซ้อน เช่น ตัวอย่างในรูปที่ 5 โฟลว์กราฟ (b) มีค่าปัจจัยความซับซ้อนสูงกว่าโฟลว์กราฟ (a) ถือเป็นปกติเพราะการวัดปัจจัยความซับซ้อนของโฟลว์กราฟ (b) เป็นการเพิ่มลำดับ



รูปที่ 5 แสดงโฟลว์กราฟที่ไม่มีโครงสร้าง D

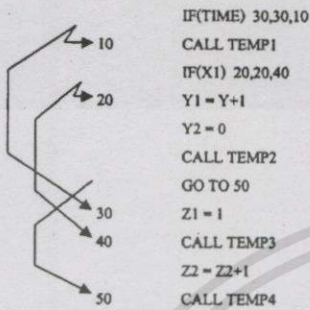
การทำงานเข้ามา ดังนั้นการวัดปัจจัยความซับซ้อนใช้ในกรณีที่โฟลว์กราฟนั้นประกอบด้วยโครงสร้าง D เท่านั้น ปัญหาที่เกิดขึ้นในการวัดปัจจัยความซับซ้อนอีกปัญหาหนึ่งคือ เมื่อมีโครงสร้างที่ไม่ใช่ D รวมอยู่ในโครงสร้างที่เป็น D ค่าไซโคลเมตริกของโครงสร้าง D จะรวมอยู่ในค่าปัจจัยความซับซ้อนทั้งหมดดังเช่นในรูปที่ 4 โครงสร้างที่ไม่ใช่ D รวมอยู่ในโครงสร้าง D_0

2.1.3 การวัดปมของโปรแกรม (The knot measure)

Knot measure เป็นการวัดจำนวนจุดทั้งหมดที่มีการข้ามบรรทัดการควบคุม (control-flow lines) วิธีการนี้สามารถที่จะคำนวณได้ดังนี้

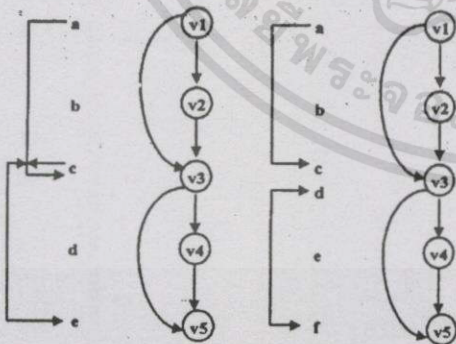
ให้ช่วงของการควบคุมการทำงานจากบรรทัด a ไปยังบรรทัด d เขียนเป็นคู่ลำดับ (a,d) โดยให้ $\min(a,d)$ คือบรรทัดเริ่มต้นการควบคุมของ (a,d) และ $\max(a,d)$ คือบรรทัดสิ้นสุดการควบคุมของ (a,d) ดังนั้น knot จะเกิดก็ต่อเมื่อ $\min(a,d) < \min(c,f) < \max(a,d)$ และ $\max(c,f) > \max(a,d)$ หรือ $\min(a,d) < \max(c,f) < \max(a,d)$ และ $\min(c,f) < \min(a,d)$

Woodward และคณะได้นำเสนอการวัดปมของโปรแกรม หรือเรียกว่า knot measure โดยนำมาใช้เป็นเกณฑ์แสดงถึงความซับซ้อนของโปรแกรม [7] ซึ่งรูปแบบการวัดของโปรแกรมควบคุมโครงสร้างนี้เป็นโปรแกรมที่มีลำดับและมีบรรทัดควบคุมลำดับการทำงานของโปรแกรมเพิ่มเข้ามา ดังรูปที่ 6 แสดงการควบคุมการไหลจากบรรทัดการทำงานของโปรแกรมภาษาฟอร์แทรน จะพบว่าจำนวนจุดตัดที่เกิดขึ้นของบรรทัดควบคุมการไหลมี 3 จุด หมายความว่าโปรแกรมนี้นับปมเกิดขึ้นทั้งหมด 3 ปม



รูปที่ 6 แสดงการควบคุมการไหลจากบรรทัดการทำงานของโปรแกรม

สรุปว่าปมของโปรแกรมจะเกิดขึ้นเมื่อมีการกระโดดออกจากขอบเขตการทำงานของ (a,c) ดังรูปที่ 7(a) วิธีการนี้สามารถนำไปประยุกต์ใช้กับโปรแกรมการไหลของกราฟได้ โดยให้ลำดับของบรรทัดที่อยู่ในช่วงการควบคุม แสดงถึง edge ของกราฟ เช่น คู่ลำดับ (a,c) ก็คือ edge (a,c) โดยที่ a เป็นโนดเริ่มต้น และ c เป็นโนดสิ้นสุด ดังนั้นการนับปมจากโพล์กราฟ พิจารณาในรูปที่ 7(a) และรูปที่ 7(b) โนดในโพล์กราฟคือ ลำดับของคำสั่งในโปรแกรมจะมีทางเข้าทางเดียวและทางออกทางเดียว โดยไม่มีการส่งข้อมูลภายใน เมื่อพิจารณารูปที่ 7(a) จะเห็นว่าบรรทัด



รูปที่ 7 (a) แสดงโพล์กราฟกับการเกิดปม (b) แสดงโพล์กราฟที่ไม่เกิดปม

ที่ c ทำ v3 และมีการส่งข้อมูลภายใน (internal transfer) จึงทำให้เกิดปมขึ้น ส่วนในรูปที่ 7(b) บรรทัดที่ c และ d จะทำ v3 เหมือนกันแต่ไม่มีการส่งข้อมูลภายในจึงไม่เกิดปมขึ้น ดังนั้นสรุปได้ว่า ปมเกิดขึ้นก็ต่อเมื่อโนดใดโนดหนึ่งมีบรรทัดการทำงาน 1 บรรทัด และมีการส่งข้อมูลภายในเกิดขึ้น

2.2 โครงสร้างการไหลของข้อมูล

ลักษณะการไหลของข้อมูลมี 2 แบบ คือ โลกอลโพล์ (local flow) และโกลบอลโพล์ (global flow) โดยนิยามได้ดังนี้ [3]

นิยาม 1 โกลบอลโพล์ของข้อมูลจากโหนด A ไปยังโหนด B หมายถึง การไหลของข้อมูลจากโหนด A ไปยังโหนด B โดยผ่านโครงสร้างข้อมูลโกลบอล D แล้ว A เก็บข้อมูลไว้ที่ D และ B เรียกใช้ข้อมูลจาก D

นิยาม 2 โลกอลโพล์ของข้อมูลจากโหนด A ไปยังโหนด B หมายถึงการไหลของข้อมูลจากโหนด A ไปยังโหนด B โดยข้อมูลมีลักษณะดังนี้

- 1) ถ้า A เรียกใช้ B หรือ
- 2) ถ้า B เรียกใช้ A และ A ส่งค่าให้กับ B แล้ว B นำค่านั้นไปใช้ หรือ
- 3) ถ้า C เรียกใช้ทั้ง A และ B แล้วส่งค่าจาก A ให้กับ B

นิยาม 3 การไหลของข้อมูลเป็นโลกอลโพล์ทางตรง (direct local flow) ของข้อมูลจากโหนด A ไปยังโหนด B เมื่อการไหลของข้อมูลเป็นลักษณะที่ 1) ในนิยาม 2

นิยาม 4 การไหลของข้อมูลเป็นโลกอลโพล์ทางอ้อม (Indirect local flow) ของข้อมูลจากโหนด A ไปยังโหนด B เมื่อการไหลของข้อมูลเป็นลักษณะที่ 2) และ 3) ในนิยาม 2

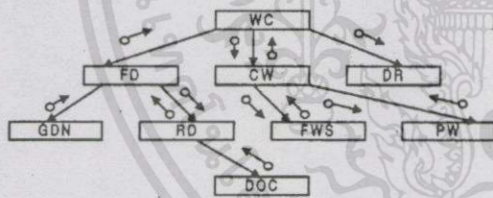
นิยาม 5 แฟนอิน (fan-in) ของโมดูล A หมายถึงจำนวนโครงสร้างข้อมูลที A รับข้อมูลเข้ามา

นิยาม 6 แฟนเอาท์ (fan-out) ของโมดูล A หมายถึงจำนวนของโครงสร้างข้อมูลที่มีการแก้ไขจาก A

Henry และ Kafura [3] ได้เสนอวิธีการวัดความซับซ้อนในการไหลของข้อมูลไว้ดังนี้

$$\begin{aligned} & \text{ความซับซ้อนในการไหลของข้อมูล(M)} \\ & = \text{length(M)} \times (\text{fan-in(M)} \times \text{fan-out(M)})^2 \end{aligned}$$

จากรูปที่ 8 เทอมของ fan-in x fan-out แสดงถึงจำนวนเส้นทาง(paths) ที่เป็นไปได้จากแหล่งข้อมูลเข้าไปยังปลายทางของข้อมูลออก การด่วงน้ำหนักของ fan-in และ fan-out ขึ้นอยู่กับการศึกษาความซับซ้อนไม่เป็นเพียงแต่ความสัมพันธ์เชิงเส้นตรงในการเชื่อมต่อกับโมดูลอื่น ๆ สำหรับการสร้างความสัมพันธ์ระหว่างโมดูลจะยุ่งยากในทางปฏิบัติ เนื่องจากระบบอื่น ๆ จะมีขนาดที่ใหญ่มา



Module	fan-in	fan-out	$[(\text{fan-in}) \times (\text{fan-out})]^2$	length	complexity
WC	2	2	16	30	480
FD	2	2	16	11	176
CW	3	3	27	40	1080
DR	1	0	0	23	0
GDN	0	1	0	14	0
RD	2	1	4	28	112
FWS	1	1	1	46	46
PW	1	1	1	29	29

รูปที่ 8 แสดงการคำนวณการวัดความซับซ้อนในการไหลของข้อมูลของ Henry และ Kafura

การวิเคราะห์การไหลของข้อมูลอาจทำได้ทันทีหลังจากขั้นตอนการออกแบบ เมื่อรายละเอียดภายนอกเสร็จสมบูรณ์แต่จะวิเคราะห์ก่อนการทำโปรแกรม (implementation)

ในส่วนต่อไปจะแสดงให้เห็นความยาวของโปรแกรม ซึ่งเป็นปัจจัยที่ยังไม่สนับสนุนการวัดความซับซ้อน ดังนั้นการประมาณค่าความยาวของโปรแกรมจึงเป็นสิ่งที่ไม่จำเป็น และในความเป็นจริงปัจจัยนี้อาจละเว้นไม่ต้องใช้ในการคำนวณก็ได้

Shepperd [6] ทำการศึกษาค่าต่าง ๆ กับการวัดที่เป็นไปได้ในการทำนาระบบที่เป็นไปตามทฤษฎีการวัดของ Henry และ Kafura และได้กำหนดจำนวนจากปัญหาตามหลักการวัด บางครั้งเกิดขึ้นจากการกำหนดกันเองจากโมดูลและการสังเกตจาก indirect flow เขาคิดว่าความยาวของโมดูลไม่น่าเป็นปัจจัยที่ส่งผลให้การไหลของข้อมูลเกิดความซับซ้อน จึงแก้ไขการวัดการไหลของข้อมูลของ Henry-Kafura สำหรับโมดูล M ใหม่ ดังนี้

$$\begin{aligned} & \text{Shepperd complexity(M)} \\ & = (\text{fan-in(M)} \times \text{fan-out(M)})^2 \end{aligned}$$

Shepperd อ้างว่าการวัดของเขามีประสิทธิภาพกว่าวิธีเดิม วิธีการวัดใหม่นี้จะกำจัดข้อมูลที่ไม่ชัดเจนและพิจารณาเฉพาะการไหลของข้อมูลอย่างเดียวเท่านั้น วิธีการวัดของ Henry-Kafura ขัดแย้งกับทฤษฎีการวัดซึ่งต้องการลักษณะของคุณสมบัติเฉพาะ Shepperd พยายามดึงเฉพาะโครงสร้างการไหลของข้อมูลให้ตรงกับทฤษฎีการวัด

การศึกษาเหตุผลจากหลักเกณฑ์ที่ Shepperd ตั้งขึ้น ทดสอบได้ว่ามีขบวนการวัดสั้นขึ้นถือเป็นการพัฒนาเวลาการคำนวณได้เร็วขึ้นและเป็นการวัดเฉพาะการไหลของข้อมูลจริง ๆ

2.3 โครงสร้างข้อมูล

ส่วนนี้จะอธิบายในเรื่องของการวัดโครงสร้างข้อมูลซึ่งการวัดนี้จะสามารถบอกเกี่ยวกับความซับซ้อนของโครงสร้างข้อมูลในซอฟต์แวร์นั้นได้ การวัดโครงสร้างของข้อมูลเหมือนกับการควบคุมการไหลของข้อมูล โดยพิจารณาการวัดที่เป็นแบบโลคอล และ โกลบอล

การวัดแบบโลคอลจะวัดผลรวมของโครงสร้างในแต่ละหน่วยข้อมูลซึ่งไม่ค่อยมีใครศึกษาการวัดแบบนี้ Elliott ใช้ทฤษฎีกราฟเข้ามาช่วยในการวิเคราะห์ และวัดคุณสมบัติจากโครงสร้างข้อมูลแต่ละตัว ซึ่งมีการแบ่งออกเป็น ส่วน ๆ เหมือนในโครงสร้างควบคุมการไหล [1] Elliott ได้กำหนดให้ prime คือ ประเภทของข้อมูล เช่น integer, character และ Boolean แล้วพิจารณาตัวดำเนินการอื่น ๆ ที่สามารถสร้างโครงสร้างข้อมูลที่ซับซ้อนกว่าได้

การวิเคราะห์ความซับซ้อนของระบบทั้งหมดจะไม่สมบูรณ์ ถ้าปราศจากการวัดโครงสร้างข้อมูล การวัดความซับซ้อนจากการควบคุมการไหลอาจไม่ได้ผล ถ้าความซับซ้อนนั้นมาซ่อนอยู่ในส่วนของโครงสร้างข้อมูล ดังตัวอย่างในรูปที่ 9 แสดงการวัดความซับซ้อนของโครงสร้างข้อมูลที่ปรากฏอยู่ในบันทึกของ R.A. Humphreys ซึ่งแสดงโปรแกรม 2 โปรแกรมที่ทำงานเหมือนกัน แต่วิธีการเขียนแตกต่างกัน โปรแกรม A มีโครงสร้างความซับซ้อนการควบคุมการไหลสูงกว่าโปรแกรม B ค่าไซโคลเมตริกที่วัดได้เท่ากับ 7 ส่วนโปรแกรม B ค่าไซโคลเมตริกเท่ากับ 1 แสดงให้เห็นถึงความแตกต่างของความซับซ้อนการควบคุมการไหลของทั้ง 2 โปรแกรม ถ้าพิจารณาในส่วนของโครงสร้างข้อมูลของโปรแกรม A กับโปรแกรม B โดยการกำหนดค่า 1 ให้กับตัวแปร

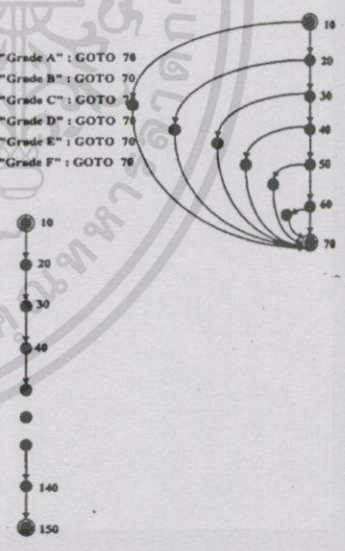
ประเภทจำนวนเต็ม (integer) หรือประเภทตัวอักษร (character) และให้ค่าเพิ่มเป็น 2 เท่าแก่ตัวดำเนินการที่อยู่ในลักษณะที่เป็นสตริง หรืออยู่ในรูปของตัวดำเนินการที่เป็นอาร์เรย์ จากโปรแกรม A พบว่ามีหน่วยข้อมูลเพียงประเภทเดียวคือ จำนวนเต็ม (integer) จึงวัดค่าความซับซ้อนโครงสร้างข้อมูลของโปรแกรม A ได้เท่ากับ 1 ส่วนโปรแกรม B มี 2 หน่วยข้อมูล หน่วยแรกเป็นจำนวนเต็มได้แก่ ตัวแปร score และ I มีค่าความซับซ้อนโครงสร้างข้อมูลเท่ากับ 1 อีกหน่วยหนึ่งวัดได้เท่ากับ 44 จากการกำหนดอาร์เรย์ 11 มิติของสายอักขระ (Character string) จึงคำนวณค่าความซับซ้อนของโครงสร้างข้อมูลนี้ ได้เท่ากับ $2 \times (2 \times (1 \times 11)) = 44$ ดังนั้นการวัดโครงสร้างข้อมูลของโปรแกรมที่แตกต่างกันเช่นในตัวอย่างของโปรแกรม A และ B ไม่สามารถวัดความซับซ้อนจากโครงสร้างควบคุมการไหลเท่านั้น

Program A

```
10 If score >= 80 Then Print "Grade A" : GOTO 76
20 If score >= 70 Then Print "Grade B" : GOTO 70
30 If score >= 60 Then Print "Grade C" : GOTO 74
40 If score >= 50 Then Print "Grade D" : GOTO 78
50 If score >= 40 Then Print "Grade E" : GOTO 70
60 If score < 40 Then Print "Grade F" : GOTO 78
70 END
```

Program B

```
10 DIM AS(11)
20 AS(10) = "Grade A"
30 AS(9) = "Grade A"
40 AS(8) = "Grade A"
50 AS(7) = "Grade B"
60 AS(6) = "Grade C"
70 AS(5) = "Grade D"
80 AS(4) = "Grade E"
90 AS(3) = "Grade F"
100 AS(2) = "Grade F"
110 AS(1) = "Grade F"
120 AS(0) = "Grade F"
130 I = Int(score / 10)
140 Print AS(I)
150 END
```



รูปที่ 9 ไพล์กราฟแสดงความแตกต่างของโครงสร้างข้อมูลและโครงสร้างควบคุมการไหล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การวัดความซับซ้อนจากขนาดของโปรแกรม

การวัดขนาดของโปรแกรมสามารถคำนวณได้ง่าย ในบทความนี้จะกล่าวถึง 2 วิธีได้แก่ การวัดจำนวนบรรทัดของโปรแกรม (line of code) และวิธีการวัดของ Halstead

โดยทั่วไปรูปแบบการวัดขนาดของโปรแกรมจะใช้วิธีการนับจำนวนบรรทัดของโปรแกรม แต่มีการกำหนดการนับที่แตกต่างกันไป บางคนนับแค่ประโยคของโปรแกรมที่มีการ execute เท่านั้น หรือนับทั้งประโยคที่มีการ execute และประโยคที่เป็นการกำหนดข้อมูล หรือบ้างก็นับทุกประโยคที่มีอยู่ในโปรแกรมรวมถึงประโยคที่เป็นประโยคอธิบายการทำงานในโปรแกรมด้วย

วิธีการของ Halstead [2] เป็นวิธีการวัดที่เป็นที่ยอมรับกันอย่างกว้างขวางทั้งในด้านอุตสาหกรรมมหาวิทยาลัย และยังสนับสนุนหลักเกณฑ์ที่ตั้งไว้ได้ด้วย การวัดของ Halstead วัดจำนวนของ operators n_1 วัดจำนวนของ operands n_2 ผลรวมของ operators N_1 และผลรวมของ operands N_2 จากพารามิเตอร์ทั้ง 4 สามารถนำมาใช้ในการคำนวณค่าความซับซ้อนจากสิ่งต่อไปนี้ได้

ความยาวของโปรแกรม (program length) คำนวณได้จากสูตร $N = N_1 + N_2$ หรือประมาณค่าได้จากสูตร

$$N' = n_1 \log_2 n_1 + n_2 \log_2 n_2$$

ปริมาตรของโปรแกรม (program volume) คำนวณจากสูตร $V = N \log_2 n$ และค่าต่ำสุดหรือ

ปริมาตรศักยภาพ (potential volume) คำนวณจากสูตร $V^* = n^* \log_2 n^*$ โดยที่ n^* คือจำนวนศัพท์ที่เป็นไปได้ศัพท์ที่เป็นไปได้ต้องการอย่างถึงฟังก์ชันที่สร้างไว้ภายในเพื่อทำงานตามที่ต้องการ ตัวอย่างเช่นโปรแกรมจัดลำดับ (sort) ในรูปที่ 10 (a) จะใช้คำสั่ง CALL SORT (X,N) (รูปที่ 10(b)) ให้โปรแกรมรูปที่ 10 (a)

ทำงาน โดยที่ X คือ อาร์เรย์ที่จัดลำดับแล้ว N คือจำนวนค่าในอาร์เรย์ ดังนั้น $n^* = 4$ ซึ่งประกอบด้วยศัพท์ที่เป็นไปได้คือ CALL, SORT(...), X และ N ระดับของโปรแกรม (program level) กำหนดเป็นอัตราส่วนระหว่างปริมาณศักยภาพกับปริมาณที่เป็นจริงซึ่งเป็นระดับของโปรแกรมในทางนามธรรมสามารถคำนวณได้จาก $L = V^*/V$ และสามารถประมาณค่าได้จากสูตร $L' = 2/n_1 \times n_2/N_2$

ความฉลาดของโปรแกรม (intelligent content) คำนวณได้จากสูตร $I = L' \times V$

จากคุณสมบัติเหล่านี้ Halstead เชื่อว่ามีผลต่อความซับซ้อนของโปรแกรม ดังนั้นถ้าเราต้องการเปรียบเทียบความซับซ้อนของโปรแกรม เราสามารถคำนวณค่าได้จากสูตรที่กำหนด แล้วนำมาเปรียบเทียบได้ทันที

4. ตัวอย่างการคำนวณความซับซ้อนของโปรแกรมด้วยวิธีต่าง ๆ

จากโปรแกรม bubble sort ในรูปที่ 10(a) สามารถแสดงการคำนวณความซับซ้อนของโปรแกรมด้วยวิธีการต่าง ๆ ได้ดังนี้

```
(a) BEGIN
      DO I = 2 TO N;
        DO J = 1 TO I;
          IF X(I) < X(J) THEN DO;
            SAVE = X(I);
            X(I) = X(J);
            X(J) = SAVE;
          END;
        END;
      END;
END;
```

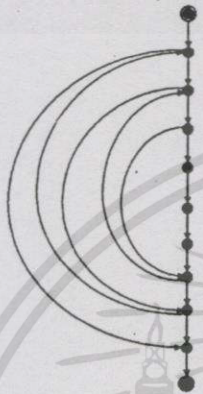
(b) CALL SORT(X,N)

รูปที่ 10 (a) แสดงโปรแกรม bubble sort

(b) แสดงศัพท์ที่เป็นไปได้ของโปรแกรม(a)

การคำนวณค่าไซโคเมตริกของ McCabe

จากโปรแกรมในรูปที่ 10 (a) เราสามารถนำมาแสดงเป็นโฟลว์กราฟได้ดังรูปที่ 11



รูปที่ 11 แสดงโฟลว์กราฟของโปรแกรม bubble sort

จากโฟลว์กราฟในรูปที่ 11 จำนวน edge (e) เท่ากับ 15 และจำนวนโนด (n) เท่ากับ 11 ดังนั้นค่าความซับซ้อนจากวิธีของ McCabe จะได้ $v(F)$ เท่ากับ $15 - 11 + 2 = 6$

การคำนวณค่าปัจจัยความซับซ้อนของ McCabe

ค่าปัจจัยความซับซ้อนของ McCabe $ev(F)$ เท่ากับ $v(F)$ คือเท่ากับ 6 ถึงแม้ว่าจะมีโครงสร้างที่เป็น D แต่มีโครงสร้างที่ไม่ใช่ D รวมอยู่ในโครงสร้าง D ดังนั้นจึงได้ค่า $ev(F)$ เท่ากับ $v(F)$ ซึ่ง McCabe ก็ได้นำเสนอถึงปัญหานี้ไว้แล้ว

การคำนวณความซับซ้อนจากปมของโปรแกรม

การวัดปมของโปรแกรมได้นำมาใช้เป็นเกณฑ์แสดงถึงความซับซ้อนของโปรแกรม พิจารณาโฟลว์กราฟรูปที่ 11 เราสามารถวัดปมของโปรแกรมได้เท่ากับ 0 เนื่องจากไม่มีการย้ายข้อมูลภายในโนดใดเลย

การคำนวณความซับซ้อนของโครงสร้างข้อมูล

จากการกำหนดให้หน่วยข้อมูลที่เป็ Integer หรือ Character มีค่าเท่ากับ 1 และหน่วยข้อมูลที่เป็น String หรือ Array มีค่าเพิ่มเป็น 2 เท่า จากโปรแกรมในรูปที่ 10(a) สมมติให้ค่าที่เก็บในอาร์เรย์เป็น Integer และ $N = 10$ จะพบว่ามี 2 หน่วยข้อมูล หน่วยข้อมูลแรกเป็น Integer ได้แก่ I, J และ SAVE ค่าความซับซ้อนเท่ากับ 1 หน่วยข้อมูลที่สองคืออาร์เรย์ที่เก็บค่า Integer ทั้งหมด N ตัว ดังนั้นหน่วยข้อมูลที่สองมีค่าความซับซ้อนเท่ากับ $2 \times (1 \times 10) = 20$

การคำนวณความซับซ้อนจากจำนวนบรรทัดของโปรแกรม

ในที่นี้เราจะนับจำนวนบรรทัดของโปรแกรมที่มีการ execute เท่านั้น จะได้เท่ากับ 11

การคำนวณความซับซ้อนของ Halstead

จากโปรแกรมในรูปที่ 10(a) และ รูปที่ 10(b) เราสามารถหาค่า n_1, n_2, N_1, N_2 และค่าที่เกี่ยวข้องตามวิธีการของ Halstead แสดงได้ดังตารางที่ 1 ถึง 5

ตารางที่ 1

แสดงการนับ operator ของโปรแกรมที่ทำงานจริง ๆ

OPERATOR	COUNT
BEGIN...END	1
;	11
DO...END	3
_	5
<	1
GOTO	2
IF...THEN	1
()	6
$n_1 = 8$	$N_1 = 30$

ตารางที่ 2

แสดงการนับ operand ของโปรแกรมที่ทำงานจริง

OPERAND	COUNT
I	5
N	1
J	4
X	6
SAVE	2
$n_2 = 5$	$N_2 = 18$

ตารางที่ 3

แสดงการนับ operator ของโปรแกรมที่เป็นไปได้

OPERATOR	COUNT
CALL	1
SORT(...)	1
$n_1^* = 2$	$N_1^* = 2$

ตารางที่ 4

แสดงการนับ operand ของโปรแกรมที่เป็นไปได้

OPERAND	COUNT
X	1
N	1
$n_2^* = 2$	$N_2^* = 2$

โดยทั่วไปทุกวันนี้การวัดขนาดของโปรแกรมเป็นวิธีที่ประสบความสำเร็จมากที่สุด สามารถแสดงให้เห็นได้อย่างชัดเจนว่าโปรแกรมที่มีขนาดใหญ่กว่าต้นทุนในการบำรุงรักษาจะสูงกว่าโปรแกรมที่มีขนาดเล็กกว่า

5. บทสรุป

การออกแบบซอฟต์แวร์จะดีหรือไม่ขึ้นอยู่กับขึ้นอยู่กับโครงสร้างภายในซึ่งการวัดคุณสมบัติภายในสามารถแสดงถึงคุณสมบัติภายนอกได้ เช่นในเรื่อง

ตารางที่ 5

แสดงค่าต่าง ๆ ของโปรแกรม bubble sort รูปที่ 10

จำนวน operators n_1	8
จำนวน operators n_2	5
ผลรวม operators N_1	30
ผลรวม operators N_2	18
จำนวนศัพท์โปรแกรม n	13
จำนวนศัพท์ที่เป็นไปได้ n^*	4
ผลรวมความยาวของโปรแกรม N	48
พยากรณ์ความยาวของโปรแกรม N'	$8 \log_2 8 + 5 \log_2 5 = 47$
ปริมาตรของโปรแกรม V	$48 \log_2 13 = 192$
ปริมาณทักษะ V^*	$4 \log_2 4 = 8$
ระดับโปรแกรม L	$8/192 = 0.042$
ระดับโปรแกรม L'	$2/8 \times 5/18 = 0.069$
ความฉลาดของโปรแกรม I	$0.069 \times 192 = 13.2$

ของ maintainability, testability, reusability และ reliability เป็นต้น วิธีการวัดความซับซ้อนแต่ละวิธีจะแสดงถึงคุณสมบัติที่แตกต่างกันไป

จากคุณสมบัติการควบคุมการไหล สามารถกำหนดค่าความซับซ้อนได้จากโพลีกราฟทันที การวัดความซับซ้อนจากโครงสร้างควบคุมการไหล McCabe ได้เสนอไว้ 2 วิธี คือ การหาค่าไซโคลเมตริกเป็นการวัดเส้นทางที่เป็นอิสระเชิงเส้นของโพลีกราฟคำนวณได้จากสูตร $v(F) = e - n + 2$ McCabe แนะนำว่าค่าไซโคลเมตริกที่เหมาะสมไม่ควรเกิน 10 ต่อมา McCabe ได้เสนอการวัดความซับซ้อน โดยพิจารณาถึงปัจจัยความซับซ้อน คือค่าไซโคลเมตริกที่ได้ลบกับจำนวนโครงสร้าง D ที่มีอยู่ในโพลีกราฟ เขาเชื่อว่า ถ้าโพลีกราฟใดมีโครงสร้าง D จะช่วยลดความซับซ้อนของโพลีกราฟได้ แต่ในการวัดปัจจัยความซับซ้อนจะใช้ได้เฉพาะกรณีที่โพลีกราฟนั้นประกอบด้วยโครงสร้าง D เท่านั้น ปัญหาที่เกิดขึ้นในการวัดด้วยวิธีนี้อีกปัญหาหนึ่งคือ เมื่อมีโครงสร้างที่ไม่ใช่ D

รวมอยู่ในโครงสร้างที่เป็น D ค่าไซโคลเมตริกของโครงสร้างที่เป็น D จะรวมอยู่ในค่าปัจจัยความซับซ้อนด้วย และวิธีสุดท้ายที่ได้นำเสนอในการวัดความซับซ้อนจากโครงสร้างควบคุมการไหลเป็นวิธีของ Woodward และคณะ ได้เสนอวิธีการวัดความซับซ้อนของโปรแกรม โดยพิจารณาจากปมที่เกิดขึ้นในโปรแกรมซึ่งเกิดเนื่องมาจากมีคำสั่งที่ต้องการย้ายการทำงานจากภายในเงื่อนไขของโปรแกรม โดยที่รูปแบบการวัดของโปรแกรมควบคุมโครงสร้างนี้จะ เป็นโปรแกรมที่มีลำดับ ก็จะต้องมีบรรทัดที่ควบคุมลำดับการทำงานของโปรแกรมด้วย

Henry และ Kafura ได้เสนอการวัดความซับซ้อนจากโครงสร้างการไหลของข้อมูลสามารถคำนวณจาก $length(M) \times (fan-in(M) \times fan-out(M))$ แต่ Shepperd เห็นว่าความยาวของโปรแกรมเป็นปัจจัยที่ไม่สนับสนุนการวัดความซับซ้อน และในการวัดการไหลของข้อมูลไม่ควรที่จะนำความยาวของโมดูลเข้ามาเกี่ยวข้องกับวงจรที่จะเป็นข้อมูลที่ได้รับการไหลของข้อมูลจริง ๆ เขาจึงกำหนดสูตรการคำนวณใหม่เป็น $(fan-in(M) \times fan-out(M))^2$ ซึ่งเป็นการวัดเฉพาะการไหลของข้อมูลจริง ๆ และยังพัฒนาเวลาในการคำนวณได้เร็วขึ้นกว่าวิธีของ Henry-Kafura

โครงสร้างข้อมูลเป็นการพิจารณาถึงตัวข้อมูลไม่ขึ้นกับโปรแกรม จะพิจารณาว่าข้อมูลถูกจัดเรียงเป็นลิสต์ กิว สแตก หรือจัดเรียงเป็นโครงสร้างแบบอื่น ๆ ดังนั้นอัลกอริทึมในการสร้าง ปรับปรุง หรือลบข้อมูลเหล่านี้จะมีการกำหนดให้ดีขึ้น โครงสร้างข้อมูลจะทำให้เราทราบถึงความยากในการเขียนโปรแกรมในการเก็บข้อมูล และความยากในการกำหนดกรณีทดสอบเพื่อพิสูจน์ว่าโปรแกรมนั้นถูกต้องหรือไม่ ในบางครั้งความซับซ้อนของ

โปรแกรมอาจเกิดจากโครงสร้างข้อมูลที่ซับซ้อนมากกว่าการควบคุมการไหลซับซ้อนหรือการไหลของข้อมูลซับซ้อน

นอกจากการวัดความซับซ้อนจากโครงสร้างของโปรแกรมแล้ว เราสามารถวัดความซับซ้อนจากขนาดของโปรแกรมได้เช่นกัน และเป็นวิธีที่ได้รับค่านิยม เนื่องจากเป็นวิธีที่ง่ายในการวัด และการเขียนโปรแกรมทำการวัด ในบทความนี้ได้นำเสนอไว้ 2 วิธี วิธีแรกได้แก่ การนับจำนวนบรรทัดของโปรแกรม ซึ่งมีหลักเกณฑ์ในการนับแตกต่างกันไปบ้างก็นับแค่ประโยคของโปรแกรมที่มีการ execute เท่านั้น หรือบ้างก็นับทุกประโยคที่มีอยู่ในโปรแกรม รวมถึงประโยคที่เป็นประโยคอธิบายการทำงานในโปรแกรม เป็นต้น วิธีที่สองเป็นวิธีของ Halstead คำนวณค่าความซับซ้อนได้จากคุณสมบัติต่าง ๆ เช่น ความยาว ปริมาตร ระดับของโปรแกรม และความฉลาดของโปรแกรม เป็นต้น โดยเขากำหนดการคำนวณออกมาเป็นสูตรง่าย ๆ สามารถคำนวณตามสูตรได้ทันที เมื่อเทียบกับวิธีแรกการคำนวณจะยุ่งยากกว่า และการเขียนโปรแกรมวัดก็ซับซ้อนกว่าด้วย ดังนั้นวิธีนับจำนวนบรรทัดของโปรแกรมจึงเป็นที่นิยมใช้กันมากกว่า

วิธีการวัดความซับซ้อนของโปรแกรมไม่ว่าจะเป็นการวัดจากโครงสร้างหรือการวัดจากขนาดของโปรแกรม เกิดจากความรู้สึกที่เกิดขึ้นในใจหรือจากการสังเกตเห็นว่าปัจจัยนั้น ๆ มีผลต่อความซับซ้อนของโปรแกรม ซึ่งค่าความซับซ้อนที่ได้จากวิธีการต่าง ๆ เหล่านี้ สามารถนำไปใช้ในการแสดงถึงผลกระทบที่มีต่อต้นทุนในการพัฒนาและการบำรุงรักษาซอฟต์แวร์ได้เหมือนกัน ซอฟต์แวร์ใดที่มีความซับซ้อนมากต้นทุนในการพัฒนาและการบำรุงรักษา ก็จะสูงด้วย ดังนั้นค่าที่ได้จากการวัดความซับซ้อนของ

โปรแกรมเราสามารถนำไปใช้ในการวางแผน และการควบคุมการจัดทำโปรแกรมหนึ่ง ๆ ได้

เอกสารอ้างอิง

- [1] Elliott, J.J., "Data complexity aspects of software", Alvey Project SE/69, PRRM/002/01, South Bank Polytechnic, London, 1988.
- [2] Halstead, M., "Elements of Software Science", Elsevier, North Holland, 1977.
- [3] Henry, S. and Kafura, D., "Software structure metrics based on information flow", *IEEE Transactions on Software Engineering*, SE-7(5), pp.510-518, 1981.
- [4] McCabe, T., "A software complexity measure," *IEEE Trans. on Software Engineering*. SE-2(4), pp. 308-20, 1976.
- [5] Norman E. Fenton and Shari Lawrence Pfleeger, *Software Metrics : A Rigorous & Practical approach*. 2nd, PWS Publishing Company, 1997.
- [6] Shepperd, M.J. and Ince, D.C., "The Use of metrics in the early detection of design errors", *Proceedings of the European Software Engineering Conference '90*, pp. 67-85, 1990.
- [7] Woodward, M.R., Hennell, M.A. and Hedley, D., "A measure of control flow complexity in program text," *IEEE Trans. on Software Engineering*, SE-5(1), pp. 45-50, 1979.

พื้นฐานทฤษฎีการวัดซอฟต์แวร์

น.ส. นิภาพร ประภาศิริ

นักศึกษาปริญญาโท สาขาวิทยาการสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

e-mail : nprapasiri@yahoo.com

ผศ.ดร. เอื้อน ปิ่นเงิน

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

e-mail : kpouen@kmitl.ac.th

บทคัดย่อ

การวัดซอฟต์แวร์จำเป็นต้องมีกฎเกณฑ์การวัดเหมือนกับการวัดทั่วไป การวัดนั้นจึงจะเป็นที่ยอมรับและถูกต้องสมบูรณ์ บทความนี้จะกล่าวถึงการวัดซอฟต์แวร์ซึ่งอาศัยระบบความสัมพันธ์ที่เกิดจากการสังเกตแล้วนำคุณสมบัติที่ได้จากการสังเกตไปแทนเงื่อนไขในรูประบบความสัมพันธ์ของจำนวน โดยแบ่งมาตรการวัดเป็น 5 ประเภท และกล่าวถึงการนำทฤษฎีการวัดไปใช้ในการกำหนดและตรวจสอบการวัดคุณสมบัติความซับซ้อนของซอฟต์แวร์ พร้อมทั้งกล่าวถึงงานวิจัยที่แสดงถึงการวัดความซับซ้อนของซอฟต์แวร์ยังไม่ประสบความสำเร็จในการแทนเงื่อนไขตามทฤษฎีการวัด และแสดงกิจกรรมการวัดซอฟต์แวร์จากคุณสมบัติต่าง ๆ พบว่าคุณสมบัติภายในของซอฟต์แวร์มีผลกระทบที่สำคัญต่อคุณสมบัติภายนอกของซอฟต์แวร์ สุดท้ายกล่าวถึงความจำเป็นในการตรวจสอบความถูกต้องสำหรับการวัดซอฟต์แวร์ เพื่อแสดงให้เห็นว่าการแทนเงื่อนไขสำหรับคุณสมบัติที่นำมาวัดนั้นใช้ได้ มีหลักและเหตุผลสนับสนุนตามทฤษฎีการวัด

Abstract

Software measurement should agree to principles of measurement, similar to measurement in other discipline, in order to gain widespread acceptance and validity. This paper present the ideas of software measurement based on empirical relation system and lead the attribute that is characterized by such a system mapping to numerical relation system and classify measurement scales as one of five major types. Later, we describe how theory of measurement can be used to define and validate measure of specifying software complexity attributes and show that the research for software complexity measures is doomed to failure mapping to measure theory and represent software measurement activities by available attributes show that internal software attributes have a significant impact on external software attributes. Finally, we conclude a necessary validating software measures and show that mapping the attributes can be used and supported by the principle of measurement.

1. บทนำ

การวัด (measurement) คือ กระบวนการกำหนดตัวเลขหรือสัญลักษณ์ให้กับคุณสมบัติ (attribute) ของ object ใด ๆ object อาจเป็น บุคคล ลักษณะของซอฟต์แวร์ หรือเหตุการณ์ต่าง ๆ เช่น การเดินทาง ขั้นตอนการทดสอบของโครงการการทำงาน ซอฟต์แวร์ คุณสมบัติ คือ ลักษณะหรือคุณสมบัติของ object เช่น ความสูง แรงดันเลือด (ของคน) ความยาว ต้นทุน หรือระยะเวลา เป็นต้น

โดยปกติการกำหนดจำนวนหรือสัญลักษณ์จะต้องเกิดจากความรูสึกที่เกิดขึ้นและการกำหนดขึ้นจากการสังเกตเกี่ยวกับคุณสมบัติของ object เช่น เมื่อเราวัดความสูงของคนคนหนึ่ง ได้ค่ามากกว่าอีกคนหนึ่ง แล้วกำหนดว่าคนที่มีความสูงมากกว่าจะเป็นคนที่สูงกว่า ค่าที่ได้จากการวัดจะแตกต่างกันไปขึ้นอยู่กับหน่วยที่เราใช้วัด เช่น นิ้ว หรือ ฟุต เป็นต้น ปัญหาที่เกิดขึ้นในการวัดคือการกำหนดโมเดลให้กับ object ที่จะวัด โมเดลนี้เป็นสิ่งที่เราจะแสดงให้เห็นถึงรายละเอียดของสิ่งที่เราสนใจซึ่งเป็นความสัมพันธ์ที่ได้จากการสังเกตเห็น เช่น ด้านวิศวกรรมซอฟต์แวร์ ต้องการโมเดลที่ดีในการวัด การวัดความยาวของโปรแกรมโดยวิธีการวัดจำนวนบรรทัดของโปรแกรม โมเดลจะต้องสามารถกำหนดบรรทัดที่จะวัดออกมาให้เห็นได้อย่างชัดเจน ในทำนองเดียวกัน การวัดต้นทุนของแรงงานที่ใช้จะต้องกำหนดโมเดลของขั้นตอนการปฏิบัติตั้งแต่เริ่มต้นจนจบ

บทความนี้ส่วนแรกเป็นการนำเสนอวิธีการที่เหมาะสมและสนับสนุนทฤษฎีการวัดซึ่งสามารถแสดงปัญหาที่เกิดขึ้นในการวัดซอฟต์แวร์และแสดงให้เห็นว่าสามารถหลีกเลี่ยงปัญหาเหล่านี้ได้ ส่วนที่ 2 นำเสนอเนื้อหาโดยสรุปของทฤษฎีการวัด ส่วนที่ 3 นำเสนอการใช้ทฤษฎีเพื่อประยุกต์กับความซับซ้อน

ของโปรแกรมวัดออกมาเป็นตัวเลขเพียงตัวเดียว ซึ่งถือเป็นพื้นฐานจากทฤษฎีการวัด และในส่วนนี้จะมีการอธิบายถึงการกำหนดลักษณะคุณสมบัติของความซับซ้อนและวิธีการวัดแบบต่าง ๆ ส่วนที่ 4 แสดงการใช้พื้นฐานการวัดอธิบายกฎเกณฑ์การวัดเพื่อวัดซอฟต์แวร์ และนำเสนอกิจกรรมการวัดซอฟต์แวร์ที่หลากหลายด้วยแนวทางที่สอดคล้องกัน พร้อมทั้งชี้ให้เห็นข้อดี ข้อเสีย ของวิธีการต่าง ๆ

2. การวัดและทฤษฎีการวัด

การวัดแบ่งออกเป็น 2 ประเภทได้แก่

- 1) การวัดคุณสมบัติโดยตรง (Direct measurement) คือ การวัดที่ไม่ขึ้นกับการวัดคุณสมบัติอื่น สามารถวัดคุณสมบัติที่สนใจได้โดยตรง เช่น การวัดความสูง และการวัดน้ำหนักของคน เป็นต้น
- 2) การวัดคุณสมบัติทางอ้อม (Indirect measurement) คือ การวัดที่เกี่ยวข้องกับคุณสมบัติอื่นด้วย อาจจะได้จากคุณสมบัติที่สามารถวัดได้โดยตรง เช่น การวัดความน่าเชื่อถือของคน เป็นต้น

ประโยชน์ของการวัดมี 3 ลักษณะคือ ใช้เพื่อการประเมิน (assessment) เพื่อการพยากรณ์ (predictive) และเพื่อการควบคุม (control) การพยากรณ์การวัดคุณสมบัติ A ขึ้นกับโมเดลทางคณิตศาสตร์ที่สัมพันธ์กับ A ของคุณสมบัติที่วัด A_1, \dots, A_n และความถูกต้องจากการพยากรณ์การวัดขึ้นกับค่าที่ได้จากการประเมินคุณสมบัติของ A_1, \dots, A_n เช่น การประมาณทรัพยากรของโครงการ (project resource) ไม่สามารถนำมาประยุกต์ใช้กับ โมเดลในการประมาณค่าของต้นทุนที่มีการกำหนดพารามิเตอร์ที่แน่นอนได้ การพยากรณ์การวัด โดยโมเดลเดียวกันไม่เพียงพอจะต้องมีการกำหนดกฎเกณฑ์เพื่อกำหนดพารามิเตอร์

ของโมเดลและเป็นตัวแปลงผลลัพธ์ ตัวอย่างเช่น ในกรณีของการพยากรณ์ความน่าเชื่อถือของซอฟต์แวร์ จะต้องใช้การประมาณค่าความเป็นไปได้สูงสุดสำหรับกำหนดพารามิเตอร์ของโมเดล และใช้สถิติของ Bayesian สำหรับเป็นตัวแปลงผลลัพธ์ โมเดลที่มีกฎเกณฑ์ทั้งสองร่วมกันเรียกว่า ระบบการพยากรณ์จากโมเดลการวัดเดียวกันนี้อาจจะให้ผลที่แตกต่างกันได้ ถ้าเราใช้กฎเกณฑ์การพยากรณ์ที่แตกต่างกัน

การกำหนดวัตถุประสงค์ประสงค์เป็นพื้นฐานสำหรับกิจกรรมการวัด ผู้วัดต้องเข้าใจวัตถุประสงค์หรือเป้าหมายของการวัดด้วย ประการแรกคือ จำเป็นที่จะต้องรู้ว่าต้องการวัดอะไร วัดเพื่ออะไร เพื่อการประเมิน เพื่อการพยากรณ์หรือเพื่อการควบคุม ต่อไปจะต้องรู้ว่าเอ็นทิตีใดบ้างเกี่ยวข้องกับสิ่งที่สนใจ แล้วทำการเลือกคุณสมบัติของเอ็นทิตีนั้นมาทำการวัด ข้อกำหนดของการวัดคือจะต้องเข้าใจรายละเอียดของเอ็นทิตีและคุณสมบัติอย่างถูกต้อง

2.1 ระบบความสัมพันธ์ที่เกิดจากการสังเกต

การวัดโดยตรงของคุณสมบัติเฉพาะวัดได้จากกลุ่มของเอ็นทิตีที่เกิดขึ้นจากความเข้าใจโดยสัญชาตญาณกับคุณสมบัตินั้น ความเข้าใจโดยสัญชาตญาณนี้จะนำมาสู่ข้อกำหนดของความสัมพันธ์ที่เกิดขึ้นจากการสังเกตเอ็นทิตีเหล่านั้น ดังนั้นการกำหนดให้กลุ่มของเอ็นทิตี C สัมพันธ์กับกลุ่มความสัมพันธ์ที่เกิดจากการสังเกตหรือเรียกว่า empirical relation system แสดงความสัมพันธ์เป็นคู่ลำดับ (C,R) เช่นคุณสมบัติความสูงของคนมี empirical relation (R) เป็น “สูง” “สูงกว่า” และ “สูงกว่ามาก” เป็นต้น

2.2 การแทนเงื่อนไข

การวัดคุณสมบัติของ empirical relation system (C,R) สามารถทำให้อยู่ในรูปของระบบความสัมพันธ์ของจำนวนหรือเรียกว่า numerical relation system แสดงด้วยสัญลักษณ์ (N,P) เมื่อ C ของ (C,R) แสดงให้อยู่ในรูปของ N ใน (N,P) และ R ของ (C,R) แสดงให้อยู่ในรูปของ P ใน (N,P) จะแสดงสัญลักษณ์ในรูปของฟังก์ชัน M เช่น ความสัมพันธ์ไบนารี $<$ แสดงให้อยู่ในรูปของระบบจำนวนได้เป็น $<$ โดยฟังก์ชัน M และเงื่อนไขที่ใช้แสดงความสัมพันธ์ของ (C,R) กับ (N,P) เป็น 2 ทิศทาง สามารถเขียนสัญลักษณ์ในการแสดงเงื่อนไขได้เป็น $x < y$ แทนด้วย $M(x) < M(y)$

ดังนั้นถ้า C เป็นกลุ่มของเอ็นทิตีทั้งหมด และ R แสดงความสัมพันธ์ “สูงกว่า” ฟังก์ชันความสูง M คือ C ซึ่งมีสมาชิกอยู่ในเซตจำนวนจริง (R) และใช้สัญลักษณ์ความสัมพันธ์มากกว่า เช่น “ $>$ ” ดังนั้นการแสดงเงื่อนไขที่บ่งบอกว่า A สูงกว่า B คือ $M(A) > M(B)$

การนำเสนอการวัดเพื่อหลีกเลี่ยงการกำหนดที่จะทำให้เข้าใจยาก จึงได้มีการกำหนด R ไว้ล่วงหน้า ซึ่งคุณสมบัติจะอยู่ภายใต้ข้อกำหนดของระบบจำนวน ปัญหานี้เป็นปัญหาหนึ่งของงานด้านการวัดซอฟต์แวร์

ตัวอย่างของคุณสมบัติที่มักพบเสมอ เช่น “ความซับซ้อน” หรือ “คุณภาพ” มีการแสดงออกมาให้เห็นเป็นตัวเลข เช่น การวัดความซับซ้อนโดยวิธีวัดค่าไซโคลเมตริกของ McCabe [20] หรือ ค่า E ของ Halstead [11] และการวัดคุณภาพจากสมการ fan-in / fan-out ของ Kafura และ Henry [15] เป็นต้น

2.3 ประเภทของมาตราวัดและความหมาย

สมมติว่าคุณสมบัติของเอ็นทีที่บางกลุ่มมีลักษณะเป็น (C,R) กำหนดค่าที่จะแสดงเงื่อนไขว่า A “สูงกว่า” B ด้วย $M(A) > M(B)$ โดยที่ฟังก์ชันการวัด M อาจเป็นนิ้ว ฟุต เซนติเมตร หรือเมตร ก็ได้ ค่าที่ได้จากการวัดคุณสมบัติเกี่ยวกับความสูงของคนสำหรับ (C,R) ปกติจะแสดงค่าที่แตกต่างกันไป ถ้า M' เป็นฟังก์ชันการวัดอีกระบบหนึ่ง เราสามารถแสดงให้เห็นความสัมพันธ์ของ M กับ M' ได้ดังนี้คือ ให้ c เป็นค่าคงที่ โดยที่ $c > 0$ เสมอ ดังนั้น $M = cM'$ โดยที่ M แสดงความสูงเป็นนิ้ว และ M' แสดงความสูงมีหน่วยเป็นเซนติเมตร จะได้ $c = 2.54$ การแปลงค่าจากมาตราหนึ่งไปเป็นค่าอีกมาตราหนึ่งเรียกว่า admissible transformation

ระดับของ admissible transformation คือ ประเภทของมาตราของคุณสมบัติหนึ่ง ซึ่งต้องอยู่ภายใต้ (C,R) ตัวอย่างเช่น ทุกครั้งที่เกิด admissible transformation จะมีค่าเป็นทวีคูณของสเกลค่า ประเภทของมาตราเรียกว่า อัตราส่วน (ratio scale) มาตราแบบอัตราส่วนเป็นมาตราของการวัดที่ละเอียด ซึ่งแสดงให้เห็นเกี่ยวกับ (C,R) ได้เป็นอย่างดี เราสามารถทำคุณสมบัติต่าง ๆ ให้อยู่ในรูปอัตราส่วนได้โดยเริ่มจากการทำความเข้าใจและให้ความหมายสำหรับการวัดคุณสมบัตินั้น รวบรวมข้อมูลวิเคราะห์ผลนำมาใช้อธิบาย พร้อมทั้งหาแนวทางใหม่ของคุณสมบัตินั้นแล้วทำการปรับปรุงแก้ไข เพื่อนำไปสู่ empirical relation ใหม่ ซึ่งเป็นการสนับสนุนความถูกต้องของการวัด สำหรับคุณสมบัติของซอฟต์แวร์หลาย ๆ คุณสมบัติอยู่ภายใต้กฎเกณฑ์ของ (C,R) แบ่งระดับของมาตราการวัดออกเป็น 5 ประเภทคือ [6]

(1) Nominal scale เป็นการกำหนดค่าของ attribute แยกเป็นกลุ่ม (classes หรือ categories) มาตราวัดประเภทนี้มีคุณสมบัติที่สำคัญคือ

- empirical relation system ประกอบด้วยกลุ่มที่แตกต่างกันและไม่มีลำดับระหว่างกลุ่ม
 - กลุ่มต่าง ๆ ที่มีอาจแทนด้วยตัวเลข หรือสัญลักษณ์ได้ แต่จะไม่มีขนาดหรือลำดับมาเกี่ยวข้อง
- ตัวอย่างการวัดความผิดพลาดของซอฟต์แวร์จะให้ผลลัพธ์ดังฟังก์ชัน $M(x)$ ซึ่งกำหนดค่าความผิดพลาดเป็น 3 กลุ่มที่แตกต่างกันและไม่มีลำดับของกลุ่มคือ

$$M(x) = \begin{cases} 101 & \text{ถ้า } x \text{ คือความผิดพลาดจากการระบุความต้องการ} \\ 2.73 & \text{ถ้า } x \text{ คือความผิดพลาดจากการออกแบบ} \\ 69 & \text{ถ้า } x \text{ คือความผิดพลาดจากการเขียนโปรแกรม} \end{cases}$$

(2) Ordinal scale เป็นข้อมูลที่เพิ่มเติมจาก nominal scale คือ ข้อมูลจะมีลำดับของกลุ่ม มาตราวัดประเภทนี้มีคุณสมบัติที่สำคัญคือ

- empirical relation system ประกอบด้วยกลุ่มที่มีการจัดลำดับของ attribute
- มีการจัดลำดับให้เป็นที่ยอมรับ
- ตัวเลขจะแทนค่าภายในกลุ่มเท่านั้น
- การบวก การลบ และการกระทำทางคณิตศาสตร์ต่าง ๆ จะไม่มีความหมาย

ตัวอย่าง การตัดเกรดของนักเรียนห้อง ก. กับห้อง ข. แสดงได้ดังฟังก์ชัน $M_g(x)$ และ $M_x(x)$

$$M_g(x) = \begin{cases} 0 & \text{ถ้า } x \text{ คือผลการเรียนไม่ผ่าน} \\ 1 & \text{ถ้า } x \text{ คือผลการเรียนพอใช้} \\ 2 & \text{ถ้า } x \text{ คือผลการเรียนดีพอใช้} \\ 3 & \text{ถ้า } x \text{ คือผลการเรียนดี} \\ 4 & \text{ถ้า } x \text{ คือผลการเรียนดีมาก} \end{cases}$$

$$M_v(x) = \begin{cases} 0 & \text{ถ้า } x \text{ คือผลการเรียนไม่ผ่าน} \\ 1 & \text{ถ้า } x \text{ คือผลการเรียนพอใช้} \\ 2 & \text{ถ้า } x \text{ คือผลการเรียนดีพอใช้} \\ 3 & \text{ถ้า } x \text{ คือผลการเรียนดี} \\ 4 & \text{ถ้า } x \text{ คือผลการเรียนดีมาก} \end{cases}$$

empirical relation system ประกอบด้วยกลุ่มที่มีการจัดลำดับ มีการจัดลำดับให้เป็นที่ยอมรับ ตัวเลขจะแทนค่าภายในกลุ่มเท่านั้น และการกระทำทางคณิตศาสตร์ต่าง ๆ จะไม่มีความหมาย

(3) Interval scale มาตรฐานนี้เป็นข้อมูลเกี่ยวกับขนาดความแตกต่างของกลุ่ม ซึ่งมีลักษณะที่สำคัญคือ

- มีการจัดลำดับเช่นเดียวกับ ordinal scale
- มีความแตกต่าง แต่ไม่เป็นอัตราส่วน คือ

สามารถรู้ความแตกต่างระหว่าง 2 กลุ่มจากข้อมูลที่มีการจัดลำดับแล้ว

- สามารถทำการบวก ลบ ได้ แต่การคูณและหารไม่มีความหมาย

ตัวอย่าง การวัดอุณหภูมิ วันนี้อุณหภูมิสูงสุดที่กรุงเทพฯ วัดได้ 24 องศาเซลเซียส อุณหภูมิสูงสุดที่เชียงใหม่วัดได้ 16 องศาเซลเซียส สามารถกล่าวได้ว่า อุณหภูมิที่กรุงเทพฯ สูงกว่าที่เชียงใหม่ 8 องศาเซลเซียส แต่ไม่สามารถกล่าวได้ว่า อุณหภูมิที่กรุงเทพฯ สูงกว่าที่เชียงใหม่ $1/3$ องศาเซลเซียส

(4) Ratio scale ส่วนใหญ่จะใช้มาตรฐานเป็นมาตรฐานในการวัด ratio scale มีลักษณะที่สำคัญคือ

- มาตรฐานเป็นมาตรฐานหนึ่งในการใช้วัดแผนที่มีการจัดลำดับ มีขนาดความแตกต่างระหว่างอันดับ และมีอัตราส่วนระหว่างอันดับ

- ค่าศูนย์แทนผลรวมของ attribute ที่ไม่มี
- การวัดแผนที่จะเริ่มจากศูนย์ และเพิ่มขึ้น

ด้วยระยะห่างที่เท่ากัน เรียกว่าหน่วยการวัด

- สามารถใช้การคำนวณทางคณิตศาสตร์ได้ทั้งหมด

ตัวอย่าง ความสูงของคน นาย ก. สูง 200 เซนติเมตร นาย ข. สูง 100 เซนติเมตร สามารถกล่าวได้ว่า นาย ก. สูงกว่า นาย ข. 100 เซนติเมตร และ นาย ก. สูงเป็นสองเท่าของ นาย ข.

(5) Absolute scale สำหรับการวัด M และ M' สามารถทำ admissible transformation ได้เพียงครั้งเดียวเท่านั้น สามารถวัดได้วิธีเดียว ดังนั้น M และ M' จะต้องเท่ากัน Absolute scale มีคุณสมบัติที่สำคัญคือ

- การวัดสำหรับ absolute scale ทำได้โดยการนับจำนวน element ที่อยู่ในกลุ่มของ entity นั้น

- attribute มีรูปแบบเป็น “จำนวนที่ปรากฏขึ้นของ x ใน entity นั้น”

ตัวอย่าง จำนวนคนที่ทำซอฟต์แวร์โครงการหนึ่ง สามารถวัดได้ทางเดียวเท่านั้นคือ โดยการนับจำนวนคน

การวัดที่มีวิธีการเดียวเท่านั้นเป็นสิ่งสำคัญที่แสดงให้เห็นถึงความแตกต่างระหว่าง ratio scale กับ absolute scale ข้อกำหนดที่เป็นรูปแบบของมาตราการวัดขึ้นอยู่กับ admissible transformations เราสามารถกำหนดได้ว่า ข้อความหนึ่ง ๆ ที่ใช้ในการวัดมีความหมายว่าอะไร โดยปกติข้อความแต่ละข้อความจะให้ความหมายของการวัด ข้อความนั้นจะเป็นจริงหรือเท็จก็ยังคงอยู่ภายใต้ admissible transformation ตัวอย่างเช่น ข้อความที่กล่าวว่า “สมชายสูงเป็นสองเท่าของสมศักดิ์” ข้อความนี้เป็นจริง(เท็จ) เมื่อเราวัดความสูงเป็นนิ้ว และจะยังคงเป็นจริง(เท็จ) เมื่อเราวัดความสูงเป็นค่าคงที่ที่มีค่าเป็นทวีคูณของหน่วยที่เป็นนิ้ว เช่น เซนติเมตร หรือ ฟุต เป็นต้น

ตารางที่ 1 แสดง admissible transformations ของมาตราการวัดแต่ละประเภท ซึ่งแสดงความสัมพันธ์ระหว่าง M และ M' มีความสัมพันธ์กันอย่างไร และแสดงตัวอย่างการวัดที่สัมพันธ์กับมาตราแต่ละประเภท

ตารางที่ 1 มาตราการวัด

ประเภทของมาตรา	admissible transformations	ตัวอย่าง
Nominal	1-1 mapping จาก M ไปยัง M'	การกำหนดชื่อของเอ็นทิตี, การแบ่งกลุ่มของเอ็นทิตี
Ordinal	การเพิ่มขึ้นของ $M(x) \geq M(y)$ ดังนั้น $M'(x) \geq M'(y)$ ด้วย	การเพิ่มขั้นของ: ความชอบ, ความแข็ง, ความพึงกัน จาก M ไป M' คือ รุนแรง, ความหนัก, คุณภาพของอากาศ, คะแนนการทดสอบ
Interval	$M' = aM + b$ ($a > 0$)	เกี่ยวกับเวลา, อุณหภูมิ (ฟาเรนไฮต์, เซลเซียส) การทดสอบความฉลาด (IQ)
Ratio	$M' = aM$ ($a > 0$)	ความแตกต่างของเวลา, ความยาว
Absolute	$M' = M$	การนับจำนวนเอ็นทิตี

ข้อสังเกตของความหมายที่สามารถกำหนดว่าประเภทของตัวปฏิบัติการที่เราสามารถวัดค่าออกมาแตกต่างกันคืออะไร เช่น ความหมายในการใช้ค่าเฉลี่ย (means) สำหรับการคำนวณค่าเฉลี่ยของกลุ่มของข้อมูลจะใช้กับ ratio scale แต่ไม่ใช้กับ ordinal scale ค่ากลาง (medians) มีความหมายสำหรับ nominal scale พื้นฐานการสังเกตสิ่งเหล่านี้ถูกมองข้ามจากการศึกษาการวัดซอฟต์แวร์ ซึ่งความผิดพลาดที่พบอยู่เสมอในการใช้ค่าเฉลี่ย (มากกว่าค่ากลาง) คือ การวัดค่าเฉลี่ยของข้อมูลเป็น ordinal เท่านั้น

3. การวัดความซับซ้อนของซอฟต์แวร์

ทฤษฎีการวัดเป็นสิ่งจำเป็นในการที่จะศึกษาการวัดความซับซ้อนของซอฟต์แวร์ ในส่วนนี้จะแสดงว่าการค้นหาสำหรับวัดอุปสงค์ทั่วไป ในการวัดค่าความซับซ้อนที่แท้จริงยังไม่ประสบความสำเร็จ แต่ว่าการวัดความซับซ้อนจะแสดงลักษณะที่เป็นจริงโดยไม่ต้องมีการพิสูจน์

3.1 การวัดความซับซ้อนทั่วไป

ในอดีตนักวิทยาศาสตร์ส่วนมากเห็นว่าลักษณะทั่วไปของความซับซ้อนจะให้ค่าจำนวนจริงเพียงตัวเดียว ซึ่งจำกัดการพิจารณาถึงลักษณะความซับซ้อนจากการควบคุมการไหล (control flow) เท่านั้น Zuse ได้นำเสนอการวัดความซับซ้อนจาก control flow ใน [33] มีการสมมติที่น้อยที่สุด ซึ่ง empirical relation system สำหรับโปรแกรมที่ซับซ้อนนำไปสู่มาตรา ordinal เป็นอย่างต่ำ เนื่องจากสมมติฐานนี้ปรากฏอยู่ในงานต่าง ๆ มากมาย

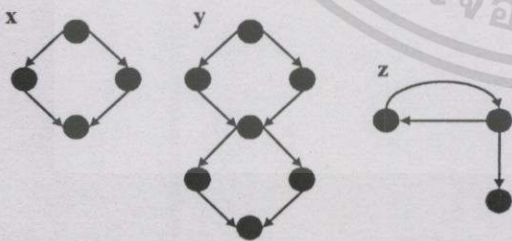
สมมติฐานที่ 1 : ให้ C คือกลุ่ม (class) ของโปรแกรมใด ๆ แล้วคุณสมบัตินี้ความซับซ้อนของการควบคุมการไหลสามารถอธิบายลักษณะจาก empirical relation system ซึ่งประกอบด้วย binary relation b ที่หมายถึง “มีความซับซ้อนน้อยกว่า” ซึ่งแสดงได้เป็น $(x,y) \in b$ ถ้า x มีความซับซ้อนน้อยกว่า y

สมมติฐานที่ 2 : รูปแบบการวัดที่เสนอ $M: C \rightarrow \mathcal{R}$ คือ การแทนความซับซ้อนซึ่งทำให้ความสัมพันธ์ b map ไปยัง $<$

สมมติฐานที่ 1 ดูแล้วพอที่จะเป็นไปได้ซึ่งจะไม่ทำในขณะที่ C เป็นคำสั่งทั้งหมดในส่วนของ b นั่นคือ สิ่งที่เห็นกันของความซับซ้อนทั่วไปซึ่งเป็นเหตุผล

ที่สอดคล้องกับคู่ลำดับของโปรแกรมที่อยู่ใน b จากตัวอย่างในรูปที่ 1 สนับสนุนเหตุผลที่ว่า $(x,y) \in b$ จากทฤษฎีการวัด แต่บางคู่ก็ไม่สามารถนำมาเปรียบเทียบกันได้ เช่น x กับ z หรือ y กับ z ถ้าถามถึงลำดับของความซับซ้อนของ x, y, z ในรูปที่ 1 ก่อนอื่นจะต้องรู้ว่าความซับซ้อนหมายถึงอะไร ก่อนที่จะตอบว่าลำดับของความซับซ้อนของ x, y, z เป็นอย่างไร เมื่อสมมติว่า b เป็นความซับซ้อนทั่วไป สิ่งเหล่านี้คงเป็นเหตุผลที่ $(x,y) \in b$ ด้วย แนวคิดในเรื่องของการนำมาเปรียบเทียบกันไม่ได้ของบางโปรแกรมในเรื่องของความซับซ้อนมีการบันทึกไว้ใน [30]

ขณะที่สมมติฐานที่ 1 พอที่จะเป็นไปได้ สมมติฐานที่ 2 สามารถตัดออกไปได้เนื่องจากสมมติฐานที่ 2 เป็นการแทนเงื่อนไข ปัญหาอยู่ที่ว่าการเปรียบเทียบกันไม่ได้ของโปรแกรม ขณะที่ b ไม่ใช่คำสั่งทั้งหมดใน C ความสัมพันธ์ $<$ เป็นคำสั่งทั้งหมดใน \mathcal{P} การวัดค่า M อาจจะพบกับคำสั่งซึ่งมีผลสะท้อนกลับมายัง C ดังนั้นจากตัวอย่าง ถ้า $M(z) < M(y)$ (ค่าไซโคลเมตริกของ McCabe ในรูปที่ 1 จะได้ $M(z) = 2$ และ $M(y) = 3$) และถ้า M คือค่าของความซับซ้อนที่วัดได้จริงแล้วในการแสดงเงื่อนไขต้องบอกได้ว่า $z < y$ ในกรณีนี้โปรแกรมมันไม่สอดคล้องกัน



รูปที่ 1 แสดงให้เห็นว่าความสัมพันธ์ของความซับซ้อนมีคุณสมบัติการถ่ายทอดหรือไม่

การวัดความซับซ้อนทั่วไป อาจจะไม่สนใจถึงทฤษฎีเลข อย่างไรก็ตามเป็นไปได้ที่จะไม่สนใจถึง

ทฤษฎีเลขในการวิจัยสำหรับการวัดสิ่งต่าง ๆ โดยเฉพาะความซับซ้อน (ซึ่งเป็นทฤษฎีการวัดที่มีอยู่จริง) เช่น คุณสมบัติเฉพาะของความซับซ้อนของโปรแกรมคือ จำนวนของเส้นทางอิสระ (independent paths) ความซับซ้อนจากค่าไซโคลเมตริกของ McCabe มีมาตรการวัดเป็น absolute ของคุณสมบัตินี้ ค่าที่วัดได้อาจเป็นมาตรา ratio จากคุณสมบัติของ “testability” กับส่วนของการทดสอบเส้นทางอิสระ คุณสมบัติเฉพาะอื่น ๆ ของความซับซ้อน เช่น ความลึกที่มากที่สุดของการรวมกลุ่ม (nesting) การกระจาย primes แยกออกเป็น tree และจำนวนเส้นทางต่าง ๆ ซึ่งทั้งหมดสามารถวัดได้จริงและวัดได้โดยอัตโนมัติ [6], [24]

Zuse ได้กล่าวถึง เรื่องของความคิดในการพิจารณาเรื่องการวัด โดยเฉพาะในเรื่องของความซับซ้อนใน [33] โดยใช้ทฤษฎีการวัดในการวิเคราะห์ถึงวิธีการวัดความซับซ้อนด้วยวิธีต่าง ๆ เขาแสดงให้เห็นว่าทฤษฎีและการสมมติเป็นสิ่งจำเป็นที่จะต้องใช้ในการวัด จากการใช้มาตรการวัดที่แตกต่างกัน ความงวมและความสัมพันธ์ของทฤษฎีการวัดซอฟต์แวร์จำนวนมาก แม้ว่าผู้เขียนไม่ได้แสดงให้เห็นความสัมพันธ์อย่างชัดเจน ต่อมา Melton et al. [21] และ Tian และ Zelkowitz [30] ได้เสนอแนวความคิดใหม่ซึ่งงานทั้ง 2 ชิ้นเป็นงานที่ผู้เขียนค้นหาลักษณะเฉพาะของความซับซ้อนเอกสารอ้างอิงที่ [30] ผู้เขียนได้เสนอค่าของความซับซ้อนที่พิจารณาถึงความเป็นจริงโดยไม่ต้องมีการพิสูจน์ (axioms) มีลักษณะเช่นเดียวกับ empirical relation สิ่งเหล่านี้หมายความว่า การแทนเงื่อนไขสามารถที่จะใช้กำหนดการยอมรับความเป็นไปได้ในการวัด หรือกล่าวได้ว่าการวัดนั้นจะมีความน่าเชื่อถือมากน้อยเพียงใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Melton et al. [21] อธิบายลักษณะเฉพาะของโปรแกรมที่ซับซ้อน โดยระบุความสัมพันธ์ของคำสั่ง \leq อย่างแน่นอนให้กับโพลีกราฟของโปรแกรมกรณีอื่น ๆ ก็กำหนด binary relation b เหมือนกับ \leq ประโยชน์ของสิ่งเหล่านี้คือให้เห็นภาพของความซับซ้อนที่ชัดเจนและเป็นการค้นหาการนำเสนอ นำไปสู่การวิเคราะห์จุดอ่อนใน [21] กล่าวว่า “การวัด M คือ ค่าที่เป็นจริงที่ทำให้ $M(x) \leq M(y)$ เมื่อใดก็ตามที่ $x \leq y$ ” สิ่งเหล่านี้ไม่สนใจเงื่อนไขที่เพียงพอของการแทนเงื่อนไข ดังนั้นขณะที่ความซับซ้อนไซโคลเมติกของ McCabe [20] จำเป็นต้องปฏิบัติให้เป็นที่พอใจ (และเพื่อเป็นการวัดในลักษณะเดียวกับ Melton et al. [21]) ไม่ได้เป็นการวัดในการแสดงถึงความรู้สึก (เมื่อรูปที่ 1 $M(z) < M(y)$ แต่ไม่ใช่กรณีที่ $z \leq y$) Tian และ Zelkowitz ใช้จุดอ่อนนี้เหมือนกันในการแทนเงื่อนไข แต่พวกเขายอมรับว่าความสัมพันธ์เป็นสิ่งจำเป็นและก่อให้เกิดความเชื่อถือได้

3.2 คุณสมบัติของ Weyuker

แม้ว่าเหตุผลที่กล่าวมาข้างต้นนั้นจะชัดเจน แต่นักวิจัยยังทำการค้นหาวิธีการวัดความซับซ้อนที่ให้ค่าจริงเพียงค่าเดียว ซึ่งนักวิจัยหวังว่าจะมีคุณสมบัติที่สามารถนำมาใช้เป็นปัจจัยของคุณสมบัติต่าง ๆ ได้ เช่น การเข้าใจ (comprehensibility) ความถูกต้อง (correctness) การทดสอบได้ (testability) และความง่ายในการสร้าง (ease of implementation) [20], [22] ค่าความซับซ้อนที่สูงแสดงถึงการทำความเข้าใจได้ยาก ความน่าเชื่อถือต่ำและอื่น ๆ บางครั้งการวัดสิ่งเหล่านี้เป็นการวัดถึงคุณภาพด้วย [15] ในกรณีที่วัดค่าต่าง ๆ ได้สูงจะแสดงถึงค่าของคุณภาพที่ต่ำ

การพยายามหาวิธีการวัดที่จะอธิบายลักษณะของคุณสมบัติที่แตกต่างกันหลาย ๆ คุณสมบัติเป็น

อันตรายเพราะว่าจะก่อให้เกิดการวัดที่มีเป้าหมายที่ขัดแย้งกัน ซึ่งเป็นการสวนทางกับการแทนด้วยทฤษฎีการวัด ไม่มีใครคิดว่าจากฟังก์ชัน M เพียงฟังก์ชันเดียวจะสามารถอธิบายลักษณะที่แสดงถึงคุณภาพของคนได้ ซึ่งอาจจะประกอบด้วยสิ่งต่าง ๆ ที่แตกต่างกันมาก เช่น การวัดความแข็งแรงทางกายภาพ (physical strength) และความฉลาด (intelligence)

ถ้ามีการวัด M เกิดขึ้น และได้ผลดังนี้

a) $M(A) > M(B)$ เมื่อ A แข็งแรงกว่า B และ

b) $M(A) > M(B)$ เมื่อ A ฉลาดกว่า B

ในความเป็นจริงแล้ว คนที่ฉลาดมากอาจมีร่างกายอ่อนแอก็ได้ แน่แน่นอนว่าไม่มีฟังก์ชัน M ใดที่สามารถวัดคุณสมบัติทั้ง 2 พร้อมกันได้ แต่กระนั้นคุณสมบัติของ Weyuker [32] แสดงให้เห็นถึงความจำเป็นในการเปรียบเทียบการวัดความซับซ้อนของซอฟต์แวร์ ตัวอย่างของคุณสมบัติทั้ง 2 ซึ่ง Weyuker ได้เสนอการวัดความซับซ้อน M เป็นไปตามคุณสมบัติดังนี้

คุณสมบัติ A : สำหรับส่วนที่เป็นโปรแกรม (program bodies) ของ P และ Q $M(P) \leq M(P;Q)$ และ $M(Q) \leq M(P;Q)$

คุณสมบัติ B : คุณสมบัตินี้อยู่ในส่วนของโปรแกรม P, Q และ R โดยที่ $M(P) = M(Q)$ และ $M(P;R) \neq M(Q;R)$

คุณสมบัติ A แสดงว่าการเพิ่มคำสั่งให้กับโปรแกรมไม่สามารถลดความซับซ้อนของโปรแกรมได้ ซึ่งสะท้อนให้เห็นถึงขนาดของโปรแกรมที่เป็นปัจจัยในการใช้วัดความซับซ้อน จากคุณสมบัติ A สามารถสรุปได้ว่า ค่า comprehensibility ค่าไม่ได้เป็นปัจจัยของความซับซ้อน เพราะเชื่อกันว่าในกรณีเดียวกันเราสามารถเข้าใจโปรแกรมง่ายยิ่งขึ้น เหมือนกับเราได้เห็นโปรแกรมนั้นมากขึ้นด้วย [30] ดังนั้นการ

วัดความซับซ้อน M จาก comprehensibility ไม่สามารถเป็นไปตามคุณสมบัติ A ซึ่งการวัดความซับซ้อน M จากขนาดจะเป็นไปตามคุณสมบัติ A

คุณสมบัติ B แสดงว่าเราสามารถพบโปรแกรม 2 โปรแกรมที่มีค่าความซับซ้อนเท่ากัน และนำแต่ละโปรแกรมรวมเข้ากับโปรแกรมที่ 3 แต่ผลของความซับซ้อนนั้นแตกต่างกันสรุปคือคุณสมบัตินี้สอดคล้องกับ comprehensibility แต่ไม่สอดคล้องกับขนาดของโปรแกรม

ดังนั้นคุณสมบัติ A และ B เป็นความสัมพันธ์ที่มีความแตกต่างกันมากและนำมาเปรียบเทียบกันไม่ได้เลยในแง่ของความซับซ้อน การวัดค่าออกมาเป็นค่าเดียวไม่สามารถเป็นไปตามคุณสมบัติทั้งสองได้ โดยการดูจากขนาดและ comprehensibility แม้ว่าเหตุผลที่กล่าวมาข้างต้นยังไม่มีที่ยอมรับอย่างเป็นทางการ แต่เมื่อเร็ว ๆ นี้ Zuse ได้พิสูจน์ว่า [34] ในการนำเสนอทฤษฎีของการวัดกับความจริงที่ไม่จำเป็นต้องพิสูจน์ของ Weyuker (Weyuker's axioms) เป็นสิ่งที่ขัดแย้งกัน เขาแสดงให้เห็นว่าขณะที่ คุณสมบัติ A ต้องการมาตรา ratio สำหรับ M อย่างชัดเจน แต่คุณสมบัติ B ไม่ต้องการมาตรา ratio

เมื่อเร็ว ๆ นี้มีบทความวิจัย [5] ซึ่งเป็นบทความวิจารณ์เกี่ยวกับทฤษฎี Weyuker's axiom Chemiavsky และ Smith ได้เขียนโปรแกรมตามหลักการ Weyuker's axioms ทั้งหมด เพื่อใช้เป็นเหตุผลสนับสนุนการวัด แต่กลับไม่สามารถนำมาใช้เป็นเหตุผลสนับสนุนความซับซ้อนได้ Chemiavsky และ Smith สรุปว่า axiomatic อาจไม่สามารถนำมาปฏิบัติได้ เพราะไม่สามารถหาเหตุผลมาสรุปสิ่งเหล่านี้ได้ แต่ Chemiavsky และ Smith ยอมรับว่าเหตุผลเหล่านั้นไม่ได้กล่าวว่า Weyuker's axioms นั้นสมบูรณ์แล้ว สิ่งที่สำคัญยิ่งกว่านั้นคือ พวกเขาบ่งชี้ว่าการ

สังเกต Weyuker ไม่ได้เสนอว่า axioms นั้นสมบูรณ์แล้ว แต่ได้เสนอเพียงแต่ว่า axiom เป็นสิ่งที่จำเป็นเท่านั้น เมื่อการวัดของ Chemiavsky และ Smith ไม่ได้เป็นการวัดของคุณสมบัติเฉพาะอย่างชัดเจน แสดงว่าเป็นการทำตามกลุ่มของ axioms ที่จำเป็นสำหรับการวัดไม่ใช่สิ่งที่สนใจทั้งหมด

ปัญหาเหล่านี้มักถูกมองข้ามจากบทเรียนเบื้องต้นของทฤษฎีการวัด คือการกำหนดให้อยู่ในระบบจำนวน (numerical mapping) จากบทความที่ [5] และ [32] ทำให้สับสนซึ่งเกิดขึ้นจากการเข้าใจผิดของหลักการทั้ง 2 และมองข้ามทฤษฎีการวัดอย่างสมบูรณ์

4. การรวมกฎเกณฑ์การวัดสำหรับการวัดซอฟต์แวร์

4.1 ประเภทของการวัดซอฟต์แวร์

กิจกรรมการวัดของซอฟต์แวร์แบ่งเอ็นทิตีออกเป็น 3 กลุ่ม

การผลิต (Processes) คือกิจกรรมที่เกี่ยวข้องกับซอฟต์แวร์ที่เกิดขึ้นตามช่วงเวลา

ผลิตภัณฑ์ (Products) คือ วัตถุที่มนุษย์ทำขึ้น (artefacts) สิ่งที่สามารถจับได้ หรือเอกสารซึ่งเกิดจากการผลิต

ทรัพยากร (Resources) คือ สิ่งที่ใช้ในการผลิต เราแบ่งคุณสมบัติของเอ็นทิตีเหล่านี้ออกเป็น คุณสมบัติภายใน (internal attributes) และคุณสมบัติภายนอก (external attributes) โดยคุณสมบัติภายในของผลิตภัณฑ์ การผลิตหรือทรัพยากร คือ สิ่งที่สามารถวัดได้จริง เช่น ความยาวเป็นคุณสมบัติภายในของเอกสารที่เกี่ยวข้องกับซอฟต์แวร์ ในขณะที่ความยาวถือเป็นคุณสมบัติภายในของการผลิตซอฟต์แวร์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติภายนอกของผลิตภัณฑ์ การผลิต หรือทรัพยากร คือ สิ่งที่สามารถวัดได้ว่า ในส่วนของผลิตภัณฑ์ การผลิต หรือทรัพยากรสัมพันธ์กับเอ็นทิตีอื่น ๆ ในภาวะแวดล้อมของตัวเองอย่างไร เช่น ความน่าเชื่อถือของโปรแกรมหนึ่ง (คุณสมบัติของผลิตภัณฑ์) ไม่ได้ขึ้นกับตัวโปรแกรมเองเท่านั้นยังขึ้นอยู่กับคอมพิวเตอร์ ความสามารถของเครื่องและผู้ใช้ด้วย ผลผลิต (Productivity) เป็นคุณสมบัติภายนอกของทรัพยากร คือ คน (อาจเป็นคนเดียวหรือหลายคน) ผลผลิตจะขึ้นอยู่กับรูปแบบของการผลิตและคุณภาพของผลผลิตที่ผลิตออกมา

ผู้จัดการซอฟต์แวร์ (software managers) และผู้ใช้ซอฟต์แวร์ (software users) ต้องการอยากให้มีการวัดและทำนายคุณสมบัติภายนอก ซึ่งคุณสมบัติภายนอกเหล่านี้จะวัดได้โดยทางอื่นเท่านั้น เช่น ผลผลิตของพนักงานส่วนมากจะวัดเป็นอัตราส่วนของปริมาณของรหัสสินค้าที่เกิดขึ้น (คุณสมบัติภายในของผลิตภัณฑ์) กับความพยายาม (effort) (คุณสมบัติภายในของการผลิต) ปัญหาการวัดผลผลิตขึ้นอยู่กับ การทำเอกสารว่าเอกสารนั้น ให้ข้อมูลดีหรือไม่ ในทำนองเดียวกันคุณภาพของระบบซอฟต์แวร์ (คุณสมบัติภายนอกของผลิตภัณฑ์) มักจะกำหนดเป็นอัตราส่วนของข้อบกพร่องที่พบระหว่างทำการทดสอบตามแบบแผน(คุณสมบัติภายในของการผลิต) กับขนาดที่วัดจาก KLOC [12] เหตุผลสำหรับผู้พัฒนา คือการวัดคุณภาพนี้ไม่สามารถกล่าวได้ว่าเป็นค่าที่วัดได้จากมุมมองของผู้ใช้งาน การศึกษา empirical มีการแนะนำสิ่งเหล่านี้อาจจะเกี่ยวข้องกันจริงเพียงเล็กน้อยระหว่างข้อบกพร่อง (faults) กับความผิดพลาด (failures) จริง ๆ ของซอฟต์แวร์ในการปฏิบัติ เช่น Adams [1] ทำการศึกษาที่สำคัญจากจำนวนของระบบของซอฟต์แวร์ใหญ่ ๆ ที่ใช้กันอยู่จากที่ต่าง ๆ

ทั่วโลก เขาพบว่าสัดส่วนข้อบกพร่องเกือบทั้งหมดไม่ได้ก่อให้เกิดความผิดพลาด ข้อบกพร่องที่ทราบอยู่แล้วเกิดขึ้นน้อยกว่า 2 % ส่วนมากเกิดจากความผิดพลาดทั่ว ๆ ไป

การตกลงร่วมกันอย่างแท้จริงเกี่ยวกับการทำให้เกิดข้อกำหนดของคุณสมบัติภายนอก ข้อบกพร่องหนึ่งคือ ข้อกำหนดของความน่าเชื่อถือของโปรแกรม ในส่วนของความน่าจะเป็นจากความผิดพลาดในการปฏิบัติที่เป็นอิสระภายในสภาพแวดล้อมหนึ่งที่กำหนดให้ [14], [19] ในกรณีนี้เราต้องการวัดคุณสมบัติภายในของการผลิต การผลิตเกิดขึ้นทุกครั้งที่ของช่วงที่มีการปฏิบัติการของซอฟต์แวร์ระหว่างการสังเกตความผิดพลาดซึ่งคุณสมบัติที่สำคัญอยู่ระหว่างการผลิต

4.2 กิจกรรมการวัดซอฟต์แวร์

ความหลากหลายที่เห็น ได้ชัดเจนในหัวข้อของ “การวัดซอฟต์แวร์ หรือ software metrics” ทั้งหมดสามารถสรุปได้ตามที่กล่าวมาข้างต้น [9] ในที่นี้เราจะนำมาเสนอกิจกรรมการวัดซอฟต์แวร์ 3 แบบดังนี้

1) โมเดลต้นทุน (Cost Modeling) เกี่ยวข้องกับการพยากรณ์คุณสมบัติของ “ความพยายาม” หรือเวลาที่ต้องการสำหรับการผลิตเพื่อพัฒนาโปรแกรม (ปกติจากขั้นตอนการรวบรวมรายละเอียดจนถึงขั้นตอนการ implement) ส่วนมากจะนำไปสู่ระบบการพยากรณ์ภายใต้โมเดลซึ่งมีสูตรดังนี้ $E = f(S)$ โดยที่ E คือ ความพยายามของคน คิดเป็นเดือน และ S คือ ขนาดของระบบ ฟังก์ชัน f อาจจะนำไปสู่คุณสมบัติของผลิตภัณฑ์อื่น ๆ (เช่นความซับซ้อนหรือความน่าเชื่อถือ) เช่นเดียวกับคุณสมบัติของการผลิตและคุณสมบัติของทรัพยากร (เช่น ประสิทธิภาพ

ของโปรแกรมเมอร์) ในกรณีของ COCOMO ของ Boehm [3] กำหนดขนาดเป็นจำนวนของการปล่อยแหล่งที่มาของคำสั่ง ซึ่งเป็นคุณสมบัติของการ implement ขั้นสุดท้ายของระบบ เมื่อมีการนำระบบการพยากรณ์มาใช้ในขั้นตอนการรวบรวมรายละเอียด (specification phase) เราจึงทำการพยากรณ์ขนาดของคุณสมบัติของผลิตภัณฑ์เพื่อเชื่อมต่อเข้ากับโมเดลหมายความว่า เราทำการแทนที่ปัญหาการพยากรณ์ที่แตกต่างกัน (การพยากรณ์ความพยายาม) ด้วยปัญหาการพยากรณ์อื่น (การพยากรณ์ขนาด) ซึ่งอาจจะไม่ได้ง่ายกว่ากันเลย สิ่งเหล่านี้ไม่ได้กล่าวถึงในเรื่องของ Albrecht [2] ซึ่งวัดขนาดของระบบด้วยจำนวนของฟังก์ชันพอยต์ (function points: FP's) สิ่งเหล่านี้คำนวณได้โดยตรงจากรายละเอียดที่มีอยู่แล้ว

2) โมเดลคุณภาพและโมเดลความน่าเชื่อถือของซอฟต์แวร์ (Software Quality Models and Reliability Models) โมเดลคุณภาพที่นิยมใช้กันมักจะกำหนดคุณภาพเป็น “factors” “criteria” และ “metrics” และนำเสนอความสัมพันธ์ของคุณภาพเหล่านี้ คุณภาพจาก factors สอดคล้องกับคุณภาพภายนอกของผลิตภัณฑ์ คุณภาพจาก criteria สอดคล้องกับคุณสมบัติภายในของผลิตภัณฑ์หรือคุณสมบัติภายในของการผลิต คุณภาพจาก metrics สอดคล้องกับการนำเสนอการวัดของคุณสมบัตินี้ใน ส่วนมากการนำเสนอความสัมพันธ์ขึ้นกับทักษะที่ได้รับอิทธิพลจากความรู้ลึกส่วนตัวล้วน ๆ ความน่าเชื่อถือคือคุณสมบัติภายนอกของผลิตภัณฑ์ที่อยู่ในระดับสูงซึ่งปรากฏอยู่ในโมเดลคุณภาพทุกโมเดล ประเภทของผลิตภัณฑ์สำหรับคุณสมบัตินี้เท่านั้นสัมพันธ์กับการปฏิบัติของซอฟต์แวร์ (executable software) โมเดลความน่าเชื่อถือเกี่ยวข้องกับการพยากรณ์ความน่าเชื่อถือของซอฟต์แวร์บนพื้นฐานของเวลาในการ

สังเกตระหว่างความผิดพลาดระหว่างการทำงาน (operation) กับการทดสอบ ดังนั้นจึงใช้คุณสมบัติภายในของการผลิตมาพยากรณ์คุณสมบัติภายนอกของผลิตภัณฑ์ ระบบการพยากรณ์ [19] ใช้ในโมเดลความน่าเชื่อถือเกิดจากโมเดลการกระจายความน่าจะเป็นกับการพิจารณาข้อสรุปทางสถิติ (เช่น การประมาณค่าความเป็นไปได้สูงสุด) สำหรับกำหนดพารามิเตอร์ของโมเดลและพิจารณาการพยากรณ์สำหรับการรวมโมเดลและการประมาณค่าพารามิเตอร์เพื่อนำมาใช้สรุปเกี่ยวกับความน่าเชื่อถือต่อไป

3) วิทยาการซอฟต์แวร์ของ Halstead (Halstead's Software Science) Halstead เสนอการวัดของคุณสมบัติภายในของโปรแกรม จาก 3 คุณสมบัติซึ่งจะสะท้อนให้เห็นถึงขนาดที่แตกต่างกันคือ ความยาว (length) ศัพท์ (vocabulary) และปริมาณ (volume) [11] (ทั้งหมดกำหนดในเทอมของ μ_1 คือจำนวนของตัวดำเนินการที่ไม่ซ้ำกัน (unique operators) μ_2 คือจำนวนตัวถูกดำเนินการที่ไม่ซ้ำกัน (unique operands) N_1 คือจำนวนตัวดำเนินการทั้งหมด (total operators) และ N_2 คือจำนวนตัวถูกดำเนินการทั้งหมด (total operands) เช่น ความยาว $N = N_1 + N_2$ แม้ว่าสิ่งเหล่านี้ดูเหมือนมีเหตุผลสนับสนุนการวัดของคุณสมบัติเฉพาะจากทฤษฎีการวัดที่ปรากฏให้เห็นจริง ผลรวมของคุณสมบัติที่แตกต่างกันนำมาใช้อธิบายในการวัดความซับซ้อนของโปรแกรม การวัดอื่น ๆ ของ Halstead เช่น E คือความพยายาม และ T คือเวลาสำหรับการทำโปรแกรมหนึ่ง ๆ P คือตัวปัญหาที่แท้จริงจากทฤษฎีการวัดที่ปรากฏให้เห็นจริงสามารถคำนวณได้จากสูตร

$$E = \frac{\mu_1 N_2 \log \mu}{2 \mu_2} \quad \text{และ} \quad T = \frac{E}{18}$$

E แทนค่าของการแบ่งความรู้สึที่ต้องการ
 เข้าใจปัญหา P' และ T แทนเวลาที่ใช้จริงในการเขียน
 P มีหน่วยเป็นวินาที สิ่งเหล่านี้ถือเป็นระบบการ
 พยากรณ์ทั่ว ๆ ไป ตัวอย่างเช่น E คือการพยากรณ์
 คุณสมบัตินของการผลิตจากการเข้าใจโปรแกรม ซึ่ง
 กล่าวไว้ในหัวข้อ 2.1 ระบบการพยากรณ์ต้องการ
 ความหมายของการกำหนดพารามิเตอร์ของโมเดล
 และการแปลผลลัพธ์ กล่าวไว้ในงานของ Halstead E
 สามารถนำไปสู่การกำหนดความหมายของค่าว่า
 ความพยายาม [6]

4.3 ความสำคัญของคุณสมบัติภายใน

จาก 4.2 เราต้องการวัดคุณสมบัติภายในเพื่อ
 สนับสนุนการวัดคุณสมบัติภายนอก ซึ่งกล่าวถึงใน
 [30] เมื่อเข้าใจคุณสมบัติภายนอกของผลิตภัณฑ์กัน
 แล้ว พบว่าความน่าเชื่อถือต้องการข้อมูลเวลาที่
 ผิดพลาดภายในระหว่างทำการทดสอบหรือระหว่าง
 การทำงาน หลายบริษัทต้องการให้มีการพยากรณ์
 คุณสมบัติภายในก่อนที่จะสร้างผลิตภัณฑ์นั้นขึ้น
 เช่น ให้รายละเอียดการออกแบบของโมดูลของ
 ซอฟต์แวร์ที่ไม่มีการทดสอบ เพื่อดูว่าเมื่อทำขึ้นมาใช้
 แล้วจะยากต่อการทดสอบหรือบำรุงรักษาหรือไม่ ถึง
 เหล่านี้เป็นตัวกระตุ้นให้เกิดการศึกษาการวัด
 คุณสมบัติภายในของผลิตภัณฑ์และทำให้เกิดการ
 ศึกษาการวัดความซับซ้อนต่อไป

เมื่อพิจารณาตัวอย่างคุณสมบัติของ modula-
 rity ของผลิตภัณฑ์ วิธีการและเทคนิคของซอฟต์แวร์
 สมัยใหม่จะขึ้นกับโครงสร้างของโมดูล ถ้าโครงสร้าง
 ของโมดูลดีแล้วซอฟต์แวร์นั้นก็ดีด้วย modularity
 จากคุณสมบัติภายในของซอฟต์แวร์มีผลกระทบที่
 สำคัญต่อคุณสมบัติภายนอกของซอฟต์แวร์ เช่น
 maintainability และ reliability แม้ว่าจากที่ศึกษาใน

[29] และ [31] เกี่ยวข้องกับความสัมพันธ์ของ
 สิ่งเหล่านี้ แต่ไม่มีการอ้างอิงหลักฐานเพียงพอที่จะ
 สนับสนุนประโยชน์ของ modularity ที่ได้กล่าวมา
 แต่ใน [29] มีการอ้างอิงหลักฐานสนับสนุนให้เห็นว่า
 modularity นี้สัมพันธ์กับ maintainability โดยแสดง
 เป็นความสัมพันธ์เชิงเส้น Pressman และคณะ [25]
 ทำการสำรวจพบว่าการยอมรับ modularity มีค่าสูงเกิน
 ไปหรือไม่ก็ต่ำเกินไป อย่างไรก็ตามปัญหาหลักที่
 ศึกษาทั้งหมดคือการขาดการวัดความถูกต้องของ
 modularity ก่อน

จากเรื่องของการแทนด้วยทฤษฎีการวัด เราจะ
 ต้องมีการกำหนดข้อตกลงเกี่ยวกับ modularity ร่วม
 กันก่อนที่จะทำการวัด empirical relation บางส่วนมี
 การกำหนดไว้ใน [10] นอกจากนี้สามารถที่จะอ่านได้
 จากหนังสือเกี่ยวกับวิศวกรรมซอฟต์แวร์ทั่ว ๆ ไป
 เช่น เชื่อกันว่าค่าเฉลี่ยของขนาดของโมดูลเดียวไม่ได้
 หมายถึง modularity ของระบบแต่จะต้องแสดงโครง
 สร้างของ โมดูลทั้งหมดเป็นระดับชั้น ซึ่งเราจะ
 พิจารณาที่จำนวนของระดับและการกระจายของโม
 ดูลแต่ละระดับ

4.4 ความถูกต้องในการวัดซอฟต์แวร์

ความถูกต้องในการวัดซอฟต์แวร์เพื่อการ
 ประเมินหรือเพื่อการควบคุม ถือเป็นภารกิจสูงให้
 เห็นว่าการแทนเงื่อนไขสำหรับคุณสมบัติที่นำมาวัดนั้นใช้
 ได้ แต่สำหรับการวัดเพื่อการพยากรณ์จะต้องระบุองค์
 ประกอบทั้งหมดของระบบการพยากรณ์ให้ชัดเจน
 และเสนอสมมติฐานที่เกี่ยวข้องก่อนที่จะทำการออก
 แบบการทดลองเพื่อตรวจสอบความถูกต้องในการวัด
 ซอฟต์แวร์

หนังสือต่าง ๆ เกี่ยวกับวิศวกรรมซอฟต์แวร์
 กล่าวถึงความจำเป็นในการตรวจสอบการวัดด้วย
 เหตุผลง่าย ๆ ไว้ภายใต้หัวข้อ “การศึกษาวิธีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบความถูกต้อง (validation studies)” ซึ่งมักจะมองข้ามไป แต่ก็มีผู้ทำการตรวจสอบไว้ใน [9] และ [23] และยังมีกรกล่าวถึงปัญหาไว้ใน [27] มีการนำเสนอการวัดเพื่อการประเมินไว้ดังนี้ การตรวจสอบความถูกต้องสำหรับการวัดซอฟต์แวร์อาจจะวัดได้จากการวัดโครงสร้างคุณสมบัติภายในของโปรแกรม การตรวจสอบความถูกต้องของการวัดโดยแสดงให้เห็นว่าการวัดที่เกิดขึ้นเกี่ยวข้องกับการวัดอื่น ๆ ที่มีอยู่ ความหมายก็คือ การเสนอการวัดที่เป็นตัวแปรอิสระในระบบการพยากรณ์เป็นหลักแต่จากการศึกษาไม่ประสบความสำเร็จในการระบุความต้องการของระบบการพยากรณ์ไว้ก่อนล่วงหน้า ผลลัพธ์มักพยายามค้นหาความสัมพันธ์ที่เกิดขึ้นกับข้อมูลซึ่งเกิดเป็นตัวแปรขึ้น หลายครั้งที่ข้อมูลทำให้เกิดโครงสร้างการวัดที่สำคัญ เช่น ใน [18] การตรวจสอบความถูกต้องของการวัดโครงสร้างแสดงให้เห็นว่าโครงสร้างเหล่านั้นสัมพันธ์กับหลักฐานการวัดเช่นเดียวกับ LOC และการวัดความซับซ้อนไซโคลเมตริกของ McCabe สิ่งที่เป็นอันตรายเกี่ยวกับความสัมพันธ์ในการวัดซอฟต์แวร์ได้นำมากล่าวถึงเป็นประเด็นหลักใน [4]

จากการค้นคว้าการวัดซอฟต์แวร์อย่างจริงจังจะไม่ยอมรับวิธีการวัดที่ไม่มีการวัดความถูกต้องจะต้องมีเหตุผลสนับสนุนว่าการวัดนั้นถูกต้อง การเปรียบเทียบไม่ได้รับการยอมรับในการนำไปใช้วัดความสูงของคนคนหนึ่ง ซึ่งไม่สามารถบอกให้ทราบถึงความฉลาดของคนคนนั้นได้ ผลลัพธ์คือว่าสามารถตรวจสอบการวัดของคุณสมบัติภายในได้ไม่ตรงกับความเป็นจริง พิจารณาตัวอย่าง Albrecht's function points [2] ซึ่งกล่าวถึง การนับฟังก์ชันที่ไม่เป็นไปตามรูปแบบ หรือ Unadjusted Function Count หรือ UFC ดูเหมือนว่าเป็นเหตุผลหนึ่งของคุณสมบัติที่

สำคัญของ functionality ในเอกสารที่รวบรวมรายละเอียด อย่างไรก็ตามเป้าหมายคือการกำหนดการวัดขนาดอย่างเดียว เป็นตัวแปรอิสระในระบบการพยากรณ์สำหรับความพยายามเป็นหลัก เนื่องจากว่าได้นำ Technical Complexity Factor (TCF) มาประยุกต์ใช้กับ UFC ถึงจำนวนของ function points (FP) ซึ่งเป็นโมเดลของระบบการพยากรณ์สำหรับความพยายาม Jeffery [13] แสดงให้เห็นว่าการปรับค่าความซับซ้อนไม่ทำให้ค่าการพยากรณ์ความพยายามเพิ่มขึ้น จากการศึกษาของเขาพบว่า UFC กับ FP ไม่มีความแตกต่างกันซึ่งทั้ง 2 คือตัวพยากรณ์ความพยายาม และให้ผลเหมือนกับที่ Kitchenham และ Kansala ได้รายงานไว้

5. บทสรุป

ความขัดแย้งต่อความเห็นของคนส่วนมากในเรื่องการวัดซอฟต์แวร์ก็เหมือนกับการวัดทั่วไป จำเป็นที่ต้องมีกฎเกณฑ์ในการวัด การวัดนั้นจึงจะเป็นที่ยอมรับและถูกต้องสมบูรณ์ ทฤษฎีการนำเสนอเกี่ยวกับการวัดกล่าวไว้ว่า การวัด คือ กระบวนการกำหนดตัวเลขหรือสัญลักษณ์ให้กับคุณสมบัติ (attribute) ของ object ใด ๆ ภายใต้กฎเกณฑ์ของ empirical relation เอ็นทิตีของซอฟต์แวร์ที่น่าสนใจแบ่งออกเป็นการผลิต ผลิตภัณฑ์ และทรัพยากร สิ่งที่เราต้องการวัดก็คือคุณสมบัติของสิ่งเหล่านี้ซึ่งอาจเป็นคุณสมบัติภายในหรือคุณสมบัติภายนอกก็ได้ แม้ว่าคุณสมบัติภายนอก เช่น ความน่าเชื่อถือของผลิตภัณฑ์ ความสม่ำเสมอในการผลิต หรือผลผลิตจากทรัพยากร สิ่งเหล่านี้เป็นส่วนหนึ่งของสิ่งที่เราสนใจวัดแต่ไม่สามารถวัดได้โดยตรง เราสามารถวัดได้โดยตรงจากคุณสมบัติภายใน ในการพยากรณ์การวัดต้องการระบบการพยากรณ์ไม่ใช่แค่โมเดลเท่านั้น แต่

ต้องการกฎเกณฑ์สำหรับการพยากรณ์เพื่อกำหนดพารามิเตอร์ของโมเดลและบอกผลของการพยากรณ์ได้ด้วย ซึ่งสิ่งเหล่านี้จะขึ้นอยู่กับความถูกต้องในการวัดในด้านการประเมิน

เราใช้ทฤษฎีการวัดแสดงถึงจุดอ่อนและจุดแข็งของงานด้านการวัดซอฟต์แวร์ ประกอบด้วย การวัดความถูกต้อง งานชิ้นนี้คำนึงถึงหลักการและเหตุผลเป็นส่วนมาก โดยการใช้องค์ประกอบที่สำคัญของทฤษฎีการวัด อย่างไรก็ตามทฤษฎีการวัดจะช่วยในการกำหนดและตรวจสอบการวัดของคุณสมบัติเฉพาะของความซับซ้อนได้

เอกสารอ้างอิง

- [1] E. Adam, "Optimizing preventive service of product," IBM Res. J, vol. 28, no. 1, pp. 2-14, 1984.
- [2] A.J. Albrecht, "Measuring application development productivity," in IBM Applic. Dev : Joint SHARE/GUIDE Symp., Monterey, CA, 1979, pp. 83-92.
- [3] B. Boehm, Software Engineering Economics. Englewood Cliffs, NJ: Prentice Hall, 1981.
- [4] R.E. Courtney and D.A. Gustafson, "Shotgun correlations in software measures," IEE Software Eng. J., vol. 8, no. 1, pp. 5-13, 1993.
- [5] J.C. Chermiavsky and C.H. Smith, "On Weyuker's axioms for software complexity measures," IEEE Trans. Software Eng., vol. 17, no. 6, pp. 636-638, June 1991.
- [6] N.E. Fenton, Software Metrics: A Rigorous Approach. London:Chapman&Hall, 1991.
- [7] _____, "When a software measure is not a measure," IEE Software Eng. J., vol. 7, no. 5, pp. 357-362, May 1992.
- [8] L. Finkelstein, "A review of the fundamental concepts of measurement," Measurement, vol. 2, no. 1, pp. 25-34, 1984.
- [9] N.E. Fenton and B.A. Kitchenham, "Validating software measures," J. Software Testing, Verification & Reliability, vol. 1, no. 2, pp. 27-42, 1991.
- [10] N.E. Fenton and A. Melton, "Deriving structurally based software measures," J. Syst. Software, vol. 12, pp. 177-187, July 1990.
- [11] M.H. Halstead, Elements of Software Science. Amsterdam: Elsevier North Holland, 1975.
- [12] J. Inglis, "Standard software quality metrics," AT&T Tech. J., vol. 65, no. 2, pp. 113-118, Feb.1985.
- [13] D.R. Jeffery, G.C. Low, and M. Barnes, "A comparison of function point counting techniques," IEEE Trans. Software Eng., vol. 19, no. 5, pp. 529- 532, Mar. 1993.
- [14] Z. Jelinski and P.B. Moranda, "Software reliability research," in Statistical Computer Performance Evaluation, W. Freiberger, Ed. New York: Academic Press, 1972, pp.465-484.
- [15] D. Kafura and S. Henry, "Software quality metrics based on interconnectivity," J. Syst. & Software, vol. 2, pp. 121-131, 1981.
- [16] D. H. Krantz, R.D. Luce, P. Suppes, and A. Tversky, Foundations of Measurement, vol. 1, New York: Academic Press, 1971.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [17] H. E. Kyburg, *Theory and Measurement*, Cambridge: Cambridge Univ. Press, 1984.
- [18] H.F. Li and W.K. Cheung, "An empirical study of software metrics," *IEEE Trans. Software Eng.*, vol. 13, no. 6, June 1987.
- [19] B. Littlewood, "Forecasting software reliability," in *Software Reliability, Modelling and Identification*, S. Bittanti, Ed. (Lecture Notes in Computer Science, vol. 341). New York: Springer-Verlag, 1988, pp. 141-209.
- [20] T.J. McCabe, "A complexity measure," *IEEE Trans. Software Eng.*, vol. SE-2, no. 4 pp. 308-320, Apr. 1976.
- [21] A.C. Melton, D.A. Gustafson, J.M. Bieman, and A. A. Baker, "Mathematical perspective of software measures research," *IEEE Software Eng. J.*, vol. 5, no. 5, pp. 246-254, 1990.
- [22] J.C. Muunson and Khoshgoftaar, "The detection of fault prone modules," *IEEE Trans. Software Eng.*, vol. 18, no. 5, pp. 423-433, May 1992.
- [23] M. Neil, "Multivariate assessment of software products," *J. Software Testing Verification and Reliability*, to appear 1992.
- [24] R.E. Prather and S.G. Giulicri, "Decomposition of flowchart schemata," *Comput. J.*, vol. 24, no.3, pp. 258-262, 1981.
- [25] R.S. Pressman, *Software Engineering : A Practitioner's Approach.*, 2nd ed, New York: McGraw-Hill Int., 1987.
- [26] F.S. Roberts, *Measurement Theory with Applications to Decision Making, Utility, and the Social Sciences*, Reading, MA: Addison Wesley, 1979.
- [27] N.F. Schneidewind, "Methodology for validating software metrics," *IEEE Trans. Software Eng.*, vol. 18, no. 5, pp. 410-422, May 1992.
- [28] P.H. Sydenham, Ed., *Handbook of Measurement Science*, vol. 1, New York: J. Wiley, 1982.
- [29] D.A. Troy and S.H. Zweben, "measuring the quality of structured design," *J.Syst. Software*, vol. 2, pp. 113-120, 1981.
- [29] D.A. Troy and S.H. Zweben, "Measuring the quality of structured design," *J. Syst. Software*, vol. 2, pp. 113-120, 1981.
- [30] J. Tian and M. V. Zelkowitz, "A formal program complexity model and its applications," *J. Syst. Software*, vol. 17, pp. 253-66, 1992.
- [31] S.N. Woodfield, H.E. Dunsmore, and V.Y. Shen, "The effects of modularisation and comments on program comprehension," in *Proc. 5th Int. Conf. Software Eng.*, 1979, pp. 213-223.
- [32] E.J. Weyuker, "Evaluating software complexity measures," *IEEE Trans. Software Eng.*, vol.14, no.9, pp. 1357-1365, Sept. 1988.
- [33] H. Zuse, *Software Complexity: Measures and Methods*. Amsterdam: de Gruyter, 1990.

- [34] _____, "Support of experimentation by measurement theory," in Experimental Software Engineering Issues (Lecture Notes in Computer Science, vol. 706), H. D. Rombach, V.R. Basili, and R. W. Selby, Eds. New York: Springer-Verlag. 1993. pp. 137-140



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวัดความซับซ้อนของซอฟต์แวร์ จากโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล

Software Complexity Measurement Based On Control Flow Structure and Control Flow Data

นิภาพร ประภาศิริ

เอือน ปิ่นเงิน

สาขาวิทยาการสารสนเทศ คณะเทคโนโลยีสารสนเทศ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

บทคัดย่อ

ความซับซ้อนของซอฟต์แวร์เป็นปัจจัยสำคัญที่ส่งผลต่อคุณภาพของซอฟต์แวร์ และเนื่องจากตัววัดความซับซ้อนของซอฟต์แวร์ทั่วไปยังไม่ประสบความสำเร็จ วัดคุณสมบัติของบทความนี้ เพื่อนำเสนอแนวทางการวัดความซับซ้อนของซอฟต์แวร์โดยพิจารณาจากรหัสต้นฉบับ ซึ่งพิจารณาคุณสมบัติจาก โครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล นำไปกำหนดเป็นโมเดลการวัดความซับซ้อน และแสดงวิธีการวัดพร้อมทั้งเปรียบเทียบผลการวัดของ โปรแกรมที่มีการทำงานเหมือนกันแต่ใช้โครงสร้างการเขียนโปรแกรมต่างกัน ซึ่งให้ผลลัพธ์ความซับซ้อนที่น่าสนใจและตรงตามความรู้สึกที่เป็นจริง การวัดความซับซ้อนจากโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหลเป็นอีกแนวทางหนึ่งที่สามารถนำไปใช้ในการตัดสินใจเพื่อเลือกใช้อัลกอริทึมที่เหมาะสมในการพัฒนาโปรแกรม

Abstract

Software complexity is a major factor that influences software quality. Most software complexity measurement methods nowadays are not fully success. The objective of this paper is to present a new software measurement method that takes program source code characteristics into account. The two characteristics, control flow structure and control flow data, are used in the proposed method. Programs having different construction but performing identical task were used to compare among various software complexity metrics. The proposed method yields satisfactory results. We conclude that our method could be used to select suitable algorithm for program development.

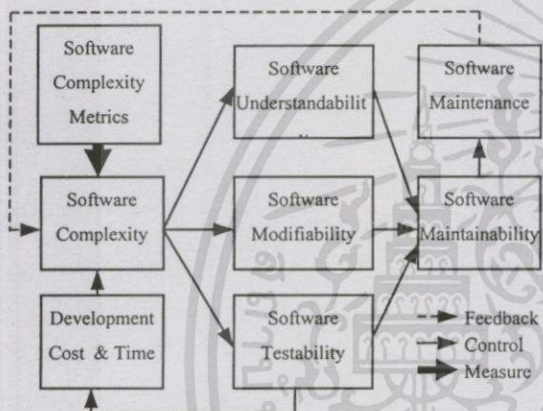
1. บทนำ

ความซับซ้อนของซอฟต์แวร์ (Software Complexity) หมายถึง ความยากต่อการเข้าใจ การแก้ไขเปลี่ยนแปลง และการทดสอบโปรแกรม [1] ความซับซ้อนของซอฟต์แวร์ถือเป็นคุณสมบัติหนึ่งของซอฟต์แวร์ที่สามารถนำไปใช้ในการประเมินคุณภาพของซอฟต์แวร์ได้ หากโปรแกรมที่เขียนขึ้นขาดคุณสมบัติที่ดี มีความซับซ้อนมาก มีโครงสร้างโปรแกรมไม่ดี ก็จะส่งผลกระทบต่อการใช้งาน การทดสอบ และการแก้ไขเปลี่ยนแปลงโปรแกรมก็จะยุ่งยากมากขึ้นด้วย ความสัมพันธ์ของความซับซ้อนของ

ซอฟต์แวร์กับกิจกรรมที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ แสดงได้ดังรูปที่ 1 [2] Fenton [3] แสดงให้เห็นว่าจากการวิจัยวิธีการวัดความซับซ้อนของซอฟต์แวร์ทั่วไปยังไม่ประสบความสำเร็จ ด้วยเหตุนี้ผู้วิจัยจึงได้ทำการศึกษาวิธีการวัดความซับซ้อนของซอฟต์แวร์จากตัววัดของ McCabe [4] ตัววัดของ Halstead [5] และ ตัววัดของ Oviedo [6] ตัววัดเหล่านี้เป็นที่รู้จักกันดี และได้รับการยอมรับจากนักวิจัยส่วนมากที่อยู่ในสาขาวิชานี้ ผู้วิจัยเปรียบเทียบข้อดีข้อเสียของแต่ละวิธีแล้วหาวิธีการที่เหมาะสม เพื่อนำเสนอตัววัดความซับซ้อนของซอฟต์แวร์ได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากข้อบกพร่องตัววัดของ McCabe เนื่องจากตัววัดของ McCabe พิจารณาเฉพาะโครงสร้างควบคุมการไหล ผู้วิจัยคิดว่าควรจะมีปัจจัยอื่นเข้ามาเกี่ยวข้องเมื่อพิจารณาจากโปรแกรมรูปที่ 2 (ก) และโปรแกรมรูปที่ 2 (ข) เป็นโปรแกรมที่ให้ผลลัพธ์การทำงานเหมือนกันแต่ใช้เงื่อนไขการเขียนโปรแกรมต่างกัน ให้ค่าความซับซ้อนจากโครงสร้างของ McCabe [4] หรือ $v(F)$ เท่ากับ 5 เหมือนกัน ตามหลักการเขียนโปรแกรมแบบโครงสร้าง การใช้เงื่อนไข Select Case นั้น เขียนโปรแกรมได้ง่ายกว่าและทำความเข้าใจได้ง่ายกว่าด้วย [7] ดังนั้นค่าความซับซ้อนของโปรแกรมรูปที่ 2 (ข) ควรจะน้อยกว่าโปรแกรมรูปที่ 2 (ก)



รูปที่ 1 ความสัมพันธ์ความซับซ้อนของซอฟต์แวร์

<pre> Sub Text1_Change() I = Val(Text1) If I = 1 Then Print "ต้องแก้ไข" ElseIf I = 2 Then Print "พอใช้" ElseIf I = 3 Then Print "ดี" ElseIf I = 4 Then Print "ดีมาก" Else Print "ใส่ตัวเลขใหม่" End If End Sub </pre>	<pre> Sub Text1_Change() I = Val(Text1) Select Case I Case 1 Print "ต้องแก้ไข" Case 2 Print "พอใช้" Case 3 Print "ดี" Case 4 Print "ดีมาก" Case Else Print "ใส่ตัวเลขใหม่" End Select End Sub </pre>
---	--

(ก)

(ข)

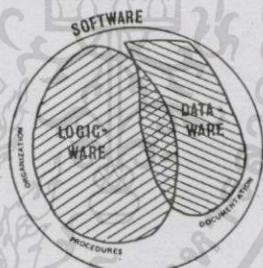
รูปที่ 2 (ก) โปรแกรมที่ใช้เงื่อนไข If...ElseIf
(ข) โปรแกรมที่ใช้เงื่อนไข Select Case

ปัจจัยที่ทำให้ความซับซ้อนจากโครงสร้างของโปรแกรมเพิ่มขึ้น คือ จำนวนตัวแปรที่ถูกใช้ตรวจสอบภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้เงื่อนไข If...EndIf For...Next Do...Loop และ Select Case...End Select ตัวแปรนี้ถือเป็นข้อมูลควบคุมการไหลจากข้อบกพร่องนี้จึงเป็นแนวทางในการนำเสนอตัววัดความซับซ้อนของโปรแกรมจากโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล ซึ่งจะกล่าวถึงข้อกำหนดของตัววัดนี้ต่อไป

บทความนี้เป็นการนำเสนอตัววัดความซับซ้อนจากโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล ซึ่งเป็นการพิจารณาคุณสมบัติ (attribute) จากระหัสต้นฉบับ (source code) และเป็นคุณสมบัติที่ได้จากองค์ประกอบหลักของซอฟต์แวร์ ตามที่ Gilb [8] ได้กล่าวไว้ องค์ประกอบหลักของซอฟต์แวร์แบ่งเป็น 2 ส่วน ได้แก่ ลอจิกแวร์ (logicware) และดาต้าแวร์ (dataware) ดังรูปที่ 3 โครงสร้างควบคุมการไหลเป็นการพิจารณาจากส่วนของลอจิกแวร์ และข้อมูลควบคุมการไหลเป็นการพิจารณาจากส่วนของดาต้าแวร์ นำไปสู่การกำหนดสูตรเพื่อวัดความซับซ้อนของโปรแกรมจากโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล



รูปที่ 3 องค์ประกอบหลักของซอฟต์แวร์

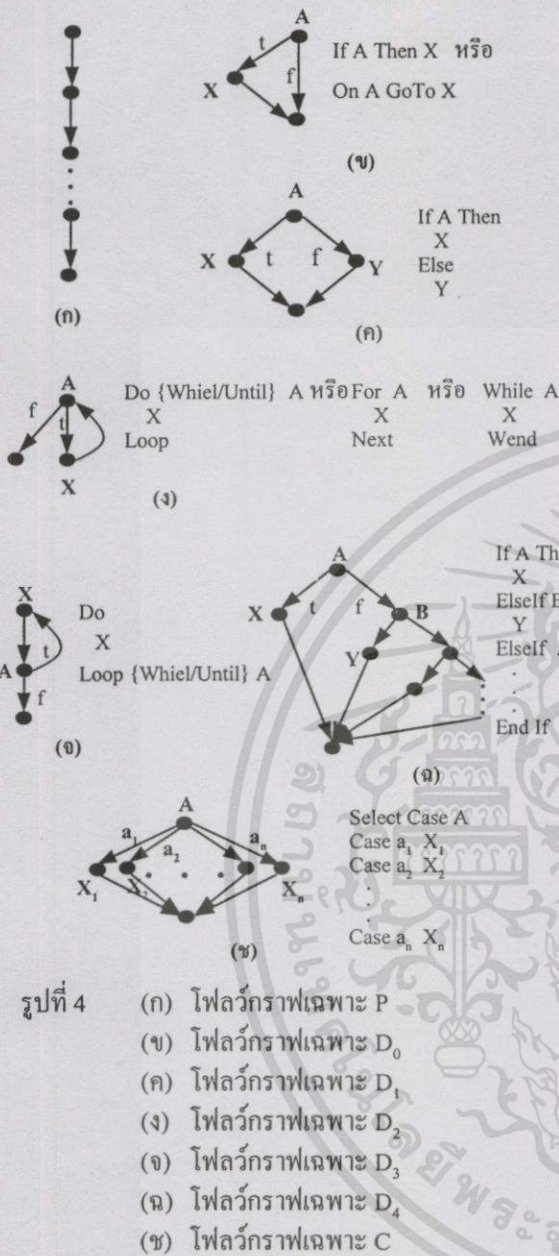
2. ปัจจัยการวัดความซับซ้อนของซอฟต์แวร์

ปัจจัยที่นำมาใช้ในการวัดความซับซ้อนของซอฟต์แวร์ ได้แก่

2.1 ความซับซ้อนจากโครงสร้างควบคุมการไหล (Control Flow Structure Complexity, CFS)

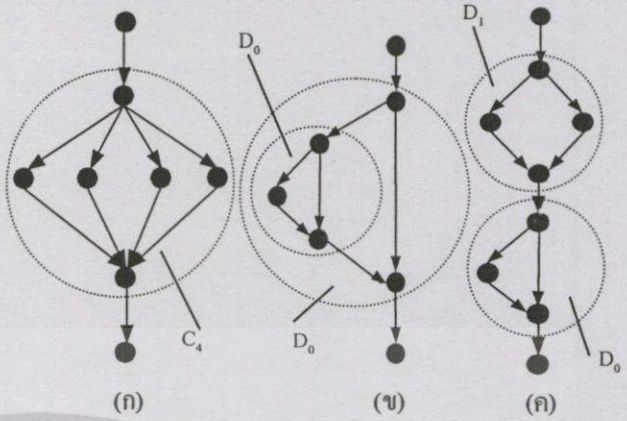
ความซับซ้อนจากโครงสร้างควบคุมการไหล คือ จำนวนโพลีกราฟย่อยที่เป็นโพลีกราฟเฉพาะ ดังรูปที่ 4 [9] แสดงโพลีกราฟเฉพาะ โครงสร้างภาษาวีซวลเบสิก

การหาค่า CFS สามารถพิจารณาได้อีกแนวทางหนึ่ง โดยพิจารณาจากจำนวนประโยคคำสั่งประเภทตัดสินใจ ได้แก่ คำสั่ง If For Do {While/Until} While Loop {While/Until} Select Case และ On...GoTo



รูปที่ 4 (ก) โพล์กราฟเฉพาะ P
(ข) โพล์กราฟเฉพาะ D_0
(ค) โพล์กราฟเฉพาะ D_1
(ง) โพล์กราฟเฉพาะ D_2
(จ) โพล์กราฟเฉพาะ D_3
(ฉ) โพล์กราฟเฉพาะ D_4
(ช) โพล์กราฟเฉพาะ C

ตัวอย่างการพิจารณาโครงสร้างควบคุมการไหลของโปรแกรม จากรูปที่ 5 (ก) มีโครงสร้างโพล์กราฟเฉพาะ C_4 และ P ค่า CFS ที่ได้มีค่าเท่ากับ 2 ส่วนรูปที่ 5 (ข) ค่า CFS จะได้เท่ากับ 3 เนื่องจากมีโครงสร้างโพล์กราฟเฉพาะ D_0 อยู่ 2 โพล์กราฟ และ P 1 โพล์กราฟ ซึ่งซ้อนใน (nesting) กันอยู่ และรูปที่ 5 (ค) ค่า CFS ได้เท่ากับ 3 เนื่องจากมีโครงสร้างโพล์กราฟเฉพาะ D_1 1 โพล์กราฟ D_0 1 โพล์กราฟ และ P 1 โพล์กราฟ จะพบว่าทุก ๆ โพล์กราฟจะต้องประกอบไปด้วยโพล์กราฟ P อยู่หนึ่งโพล์กราฟเสมอ



รูปที่ 5 (ก) โครงสร้างโปรแกรมที่ประกอบด้วยโพล์กราฟเฉพาะ C_4
(ข) โครงสร้างโปรแกรมที่ประกอบด้วยโพล์กราฟเฉพาะ D_0 ซึ่งซ้อนในกัน
(ค) โครงสร้างโปรแกรมที่ประกอบด้วยโพล์กราฟเฉพาะ D_1 เรียงลำดับ D_0

2.2 ความซับซ้อนจากข้อมูลควบคุมการไหล (Control Flow Data Complexity, CFD)

ความซับซ้อนจากข้อมูลควบคุมการไหล คือ จำนวนตัวแปรที่ใช้ตรวจสอบภายใต้เงื่อนไข รวมถึงฟังก์ชันที่มีการเรียกใช้ภายใต้เงื่อนไขด้วย เช่น

If A = B And C = D

ตัวแปรที่ควบคุมการไหล ได้แก่ ตัวแปร A และ C ดังนั้น CFD = 2

Do While (A Or B) And C

ตัวแปรที่ควบคุมการไหล ได้แก่ ตัวแปร A B และ C ดังนั้น CFD = 3

If IsDate(A) Then

ตัวแปรที่ควบคุมการไหล ได้แก่ ตัวแปร A และฟังก์ชันควบคุมการไหล ได้แก่ IsDate ดังนั้น CFD = 2

จากข้อกำหนดในการพิจารณาความซับซ้อนจากโครงสร้างควบคุมการไหล และข้อมูลควบคุมการไหล นำไปใช้ในการกำหนดโมเดลวัดความซับซ้อนซึ่งจะกล่าวถึงในหัวข้อต่อไป

3. โมเดลวัดความซับซ้อนจากโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล

จากข้อบกพร่องตัววัดของ McCabe จากโปรแกรมรูปที่ 2 (ก) และ โปรแกรมรูปที่ 2 (ข) ในการเลือกใช้โครงสร้างการเขียนโปรแกรมแตกต่างกันดังที่กล่าวมาข้างต้น เนื่องจากพิจารณาความซับซ้อนเฉพาะโครงสร้างเพียงอย่างเดียว ผู้วิจัยจึงได้นำเสนอตัววัดใหม่ซึ่งพิจารณาทั้งจากโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล เรียกว่า ตัววัดโครงสร้างควบคุมการไหลและข้อมูลควบคุมการไหล (Control Flow Structure and Control Flow Data Metric, SDM)

เนื่องจาก คาร์ด (Card) และ กลาส (Glass) [10] ได้กำหนดความซับซ้อนของซอฟต์แวร์ เท่ากับผลรวมความซับซ้อนของโครงสร้างกับความซับซ้อนของข้อมูล (structural complexity + data complexity)

ดังนั้นผู้วิจัยจึงกำหนดให้ CFS แทนค่าความซับซ้อนจากโครงสร้างควบคุมการไหล และกำหนดให้ CFD แทนค่าความซับซ้อนจากข้อมูลควบคุมการไหล และกำหนดสูตรหาค่าความซับซ้อนตัววัด SDM ได้ดังนี้

$$SDM = CFS + CFD \quad (1)$$

เนื่องจาก CFS ของทุกโปรแกรมจะประกอบด้วย โพลีกราฟเฉพาะ P อย่างน้อย 1 โครงสร้าง ดังนั้นจึงกำหนดค่าความซับซ้อนตัววัด SDM ใหม่ดังนี้

$$SDM = CFS + CFD + 1 \quad (2)$$

โดยที่ CFS คือ จำนวนโพลีกราฟเฉพาะที่ไม่ใช่โพลีกราฟ P

CFD คือ จำนวนตัวแปรและฟังก์ชันที่ใช้ตรวจสอบภายใต้เงื่อนไข

4. การทดลอง

ในส่วนนี้จะแสดงการทดลองการวัดความซับซ้อนของโปรแกรมที่เขียนด้วยภาษาวิซวลเบสิกจากตัววัด SDM

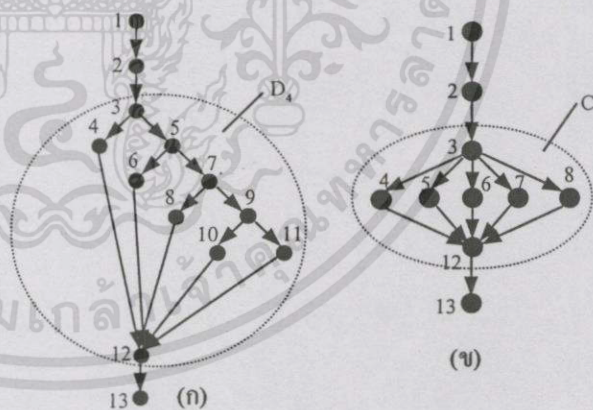
โปรแกรมทดสอบที่ 1 โปรแกรมรูปที่ 2 (ก) แสดงโครงสร้างควบคุมการไหลได้ดังรูปที่ 6 (ก) มีโครงสร้างโพลีกราฟเฉพาะ 1 โครงสร้างเป็นโครงสร้าง If... ElseIf หรือเรียกว่าโพลีกราฟเฉพาะ D_4 ดังนั้น $CFS = 1$ ส่วนจำนวนตัวแปรควบคุมการไหลหรือ CFD แสดงได้ดังตารางที่ 1 จากตารางที่ 1 แสดงจำนวนตัวแปรควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมรูปที่ 2 (ก) โหนดที่ 3

โหนดที่ 5 โหนดที่ 7 และโหนดที่ 9 มีตัวแปร 1 ควบคุมการไหลของโพลีกราฟ โหนดละ 1 ตัว ดังนั้นโพลีกราฟนี้มีตัวแปรควบคุมการไหลทั้งหมด 4 ตัว $CFD = 4$ เมื่อแทนค่า CFS และ CFD ในสูตรวัดความซับซ้อนจากตัววัด SDM จะได้ $SDM = 1 + 4 + 1 = 6$ ดังนั้นโปรแกรมรูปที่ 2 (ก) มีความซับซ้อนเท่ากับ 6

ตารางที่ 1 จำนวนตัวแปรควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมรูปที่ 2 (ก)

โหนด	ตัวแปรควบคุมการไหล	CFD _i
3	I	1
5	I	1
7	I	1
9	I	1
CFD		4

โปรแกรมทดสอบที่ 2 โปรแกรมรูปที่ 2 (ข) แสดงโครงสร้างควบคุมการไหลได้ดังรูปที่ 6 (ข) มีโครงสร้างโพลีกราฟเฉพาะ 1 โครงสร้างเป็นโครงสร้าง Select Case หรือเรียกว่าโพลีกราฟเฉพาะ C ดังนั้น $CFS = 1$ ส่วนจำนวนตัวแปรควบคุมการไหลหรือ CFD แสดงได้ดังตารางที่ 2



รูปที่ 6 (ก) โพลีกราฟเฉพาะ โปรแกรมรูปที่ 2 (ก)

(ข) โพลีกราฟเฉพาะ โปรแกรมรูปที่ 2 (ข)

จากตารางที่ 2 แสดงจำนวนตัวแปรควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมรูปที่ 2 (ข) โหนดที่ 3 มีตัวแปร 1 ควบคุมการไหลของโพลีกราฟเพียงตัวเดียว ดังนั้น $CFD = 1$ เมื่อแทนค่า CFS และ CFD ในสูตรวัดความซับซ้อนจากตัววัด SDM จะได้ $SDM = 1 + 1 + 1 = 3$ ดังนั้นความซับซ้อนของโปรแกรมรูปที่ 2 (ข) เท่ากับ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2 จำนวนตัวแปรควบคุมการไหลในโหนดเงื่อนไขของโปรแกรมรูปที่ 2 (ข)

โหนด	ตัวแปรควบคุมการไหล	CFD _i
3	I	1
	CFD	1

จากผลการวัดความซับซ้อนของโปรแกรมรูปที่ 2 (ก) และโปรแกรมรูปที่ 2 (ข) ซึ่งทั้งสองโปรแกรมมีผลลัพธ์การทำงานเหมือนกันแต่โครงสร้างการเขียนโปรแกรมต่างกัน ตามหลักการเขียนโปรแกรมแล้วการใช้โครงสร้างคำสั่ง Select Case มีความซับซ้อนน้อยกว่าการใช้โครงสร้างคำสั่ง If...Elseif ซึ่งผลการวัดจากตัววัด SDM ให้ผลความซับซ้อนตามความรู้สึกที่เป็นจริง คือ โปรแกรมรูปที่ 2 (ก) ซับซ้อนกว่าโปรแกรมรูปที่ 2 (ข)

จากตัววัด SDM นี้สามารถแก้ไขข้อบกพร่องตัววัดของ McCabe ซึ่งพิจารณาเฉพาะโครงสร้างของโปรแกรมตัววัดของ Halstead [5] ที่พิจารณาเฉพาะขนาดของโปรแกรม และแก้ไขข้อบกพร่องตัววัดของ Oviedo [6] ซึ่งประสิทธิภาพของตัววัดนี้จะลดลง ถ้าโปรแกรมนั้นมีการใช้ตัวแปรแบบโกลบอล แต่ตัววัด SDM ไม่ได้สนใจเฉพาะโครงสร้างหรือขนาดของโปรแกรมนั้นและค่าความซับซ้อนไม่ได้ขึ้นอยู่กับตัวแปรแบบโกลบอลหรือโลคอล จึงทำให้ผลการวัดตรงตามความรู้สึกที่เป็นจริงมากขึ้น ซึ่งผู้วิจัยจะแสดงการเปรียบเทียบผลการวัดความซับซ้อนจากตัววัด SDM กับตัววัดอื่น ๆ ในส่วนของผลการทดลอง

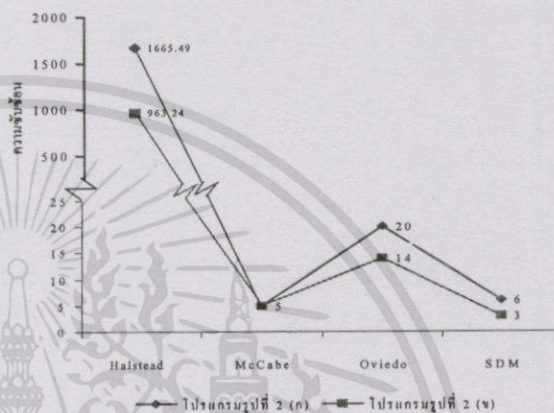
5. ผลทดลอง

ผลการวัดค่าความซับซ้อนของโปรแกรมรูปที่ 2 (ก) และโปรแกรมรูปที่ 2 (ข) ซึ่งมีผลลัพธ์การทำงานเหมือนกันแต่โครงสร้างการเขียนโปรแกรมต่างกัน แสดงการเปรียบเทียบความซับซ้อนจากตัววัดของ Halstead [5] ตัววัดของ McCabe [4] ตัววัดของ Oviedo [6] และ ตัววัด SDM ได้ดังตารางที่ 3 จากตารางที่ 3 แสดงกราฟเปรียบเทียบค่าความซับซ้อนจากตัววัดทั้ง 4 แบบได้ดังรูปที่ 7

จากกราฟรูปที่ 7 ค่าความซับซ้อนโปรแกรมรูปที่ 2 (ข) จากตัววัดทุกตัวน้อยกว่าโปรแกรมรูปที่ 2 (ก) ยกเว้นตัววัดของ McCabe ค่าความซับซ้อนของทั้งสองโปรแกรมเท่ากัน จากตัวอย่างนี้แสดงให้เห็นว่าตัววัดของ McCabe ยังมีข้อบกพร่อง ในกรณีที่มีการใช้คำสั่งเงื่อนไข Select Case ส่วนตัววัด SDM ยังคงให้ผลลัพธ์ในทำนองเดียวกับตัววัดอื่น ๆ

ตารางที่ 3 เปรียบเทียบความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และตัววัด SDM โปรแกรมรูปที่ 2 (ก) กับโปรแกรมรูปที่ 2 (ข)

ตัววัด	ความซับซ้อนโปรแกรมรูปที่ 2 (ก)	ความซับซ้อนโปรแกรมรูปที่ 2 (ข)
Halstead	1665.49	963.24
McCabe	5	5
Oviedo	20	14
SDM	6	3



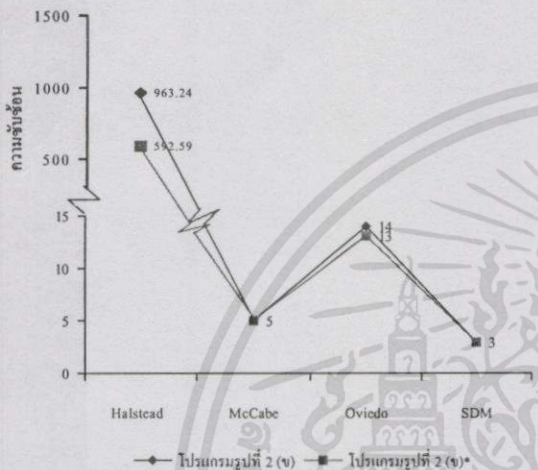
รูปที่ 7 กราฟเปรียบเทียบค่าความซับซ้อนของโปรแกรมรูปที่ 2 (ก) และ โปรแกรมรูปที่ 2 (ข)

ตัววัดของ Oviedo [6] ยังมีข้อบกพร่องเช่นเดียวกันซึ่งตัววัด SDM และตัววัดของ Oviedo จัดอยู่ในกลุ่มของตัววัดแบบเดียวกันคือ ตัววัดทั้งสองพิจารณาค่าความซับซ้อนจากโครงสร้างควบคุมการไหลและข้อมูล จากโปรแกรมรูปที่ 2 (ข) ถ้าตัดคำสั่ง I = Val(Text1) (โปรแกรมรูปที่ 2 (ข)*) แสดงการเปรียบเทียบความซับซ้อนจากตัววัดของ Halstead [5] ตัววัดของ McCabe [4] ตัววัดของ Oviedo [6] และ ตัววัด SDM ได้ดังตารางที่ 4 และแสดงกราฟเปรียบเทียบค่าความซับซ้อนจากตัววัดทั้ง 4 แบบได้ดังรูปที่ 8 จากกราฟรูปที่ 8 จะเห็นตัววัดของ Oviedo และตัววัดของ Halstead ค่าความซับซ้อนของโปรแกรมรูปที่ 2 (ข) มีค่ามากกว่าโปรแกรมรูปที่ 2 (ข)* ส่วนตัววัดของ McCabe และตัววัด SDM ให้ค่าความซับซ้อนของทั้งสองโปรแกรมเท่ากัน ผู้วิจัยคิดว่าการกำหนดค่าให้กับตัวแปร I จากภายในหรือภายนอกโมดูลที่พิจารณาไม่ส่งผลให้ค่าความซับซ้อนเพิ่มขึ้นหรือลดลง ดังนั้นผลการวัดความซับซ้อนจากตัววัด SDM จึงให้ผลที่ตรงตามความเป็นจริงมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4 เปรียบเทียบความซับซ้อนจากตัววัดของ Halstead ตัววัดของ McCabe ตัววัดของ Oviedo และ ตัววัด SDM โปรแกรมรูปที่ 2 (ข) กับ โปรแกรมรูปที่ 2(ข)*

ตัววัด	ความซับซ้อน โปรแกรมรูปที่ 2 (ข)	ความซับซ้อน โปรแกรมรูปที่ 2(ข)*
Halstead	963.24	592.59
McCabe	5	5
Oviedo	14	13
SDM	3	3



รูปที่ 8 กราฟเปรียบเทียบค่าความซับซ้อนของโปรแกรมรูปที่ 2 (ข) และโปรแกรมรูปที่ 2 (ข)*

จากผลการทดลองที่ได้พบว่าตัววัดของ McCabe ตัววัดของ Halstead และตัววัดของ Oviedo ยังมีข้อบกพร่องอยู่ จึงให้ค่าความซับซ้อนที่ไม่ตรงตามความเป็นจริง ส่วนตัววัด SDM ให้ผลลัพธ์เป็นที่น่าพอใจและตรงตามความเป็นจริง (intuitive notion) และจากการทดสอบโปรแกรมที่พัฒนาด้วยภาษาวิซวลเบสิก 1,064 โมดูล โดยเปรียบเทียบกับเกณฑ์ความซับซ้อนที่เหมาะสมของ McCabe มีค่าไม่ควรเกิน 10 ดังนั้นโปรแกรมที่มีความซับซ้อนของ McCabe (v(F)) เกิน 10 โปรแกรมที่มีความซับซ้อนมากเกินไป พบว่าค่าความซับซ้อนจากตัววัด SDM ที่มี v(F) เกิน 10 มีค่ามากกว่า 15 อยู่ร้อยละ 93.48 ดังนั้นจึงได้เสนอเกณฑ์ความซับซ้อนที่เหมาะสมสำหรับตัววัด SDM ไม่ควรเกิน 15 ถ้าโปรแกรมใดมีความซับซ้อนเกิน 15 ควรได้รับการแก้ไข [11]

6. บทสรุป

บทความนี้ได้นำเสนอตัววัดความซับซ้อนจากโครงสร้างควบคุมการไหล (CFS) และข้อมูลควบคุมการไหล (CFD) โดยพิจารณาจากรหัสต้นฉบับ (source code) ซึ่งตัววัดนี้เป็นอีกแนวคิดหนึ่งในการวัดความซับซ้อนของ

ซอฟต์แวร์โดยพิจารณาองค์ประกอบหลักทั้ง 2 ส่วนของซอฟต์แวร์ คือพิจารณาทั้งลอจิกแวลว์และค่าตัวแปร ซึ่งให้ผลลัพธ์เป็นที่น่าพอใจตรงตามความรู้สึกที่เป็นจริงมากขึ้น จึงกล่าวได้ว่าความซับซ้อนของโปรแกรมไม่ได้ขึ้นอยู่กับโครงสร้างหรือข้อมูลเพียงอย่างเดียวเท่านั้น ควรพิจารณาโครงสร้างและข้อมูลประกอบกัน ตัววัด SDM ที่นำเสนอนี้จึงน่าที่จะดีกว่าตัววัดที่พิจารณาเฉพาะโครงสร้างหรือข้อมูลเพียงอย่างเดียว ตัววัดนี้แก้ไขข้อบกพร่องจากตัววัดของ McCabe [4] Halstead [5] และ Oviedo [6] ได้ และค่าความซับซ้อนที่เหมาะสมควรมีค่าไม่เกิน 15

7. เอกสารอ้างอิง

- [1] เอื้อน ปิ่นเงิน, “ความซับซ้อนของโปรแกรม,” บิซิเนสคอมพิวเตอร์แมกซีน, ปีที่ 4, หน้า 159-162, มีนาคม 2535.
- [2] H.F. LI and W.K. CHEUNG, “An Empirical Study of Software Metrics,” IEEE Trans. on Software Engineering, Vol. SE-13, No. 6, pp. 697-708, June 1987.
- [3] N. Fenton, “Software Measurement : A Necessary Scientific Basis,” IEEE Trans. on Software Engineering, Vol. 20, No. 3, pp. 199–206, March 1994.
- [4] T.J. McCabe, “A Complexity Measure,” Software Engineering Metrics I: Measure and Validations, Vol. 1, pp. 22-51, 1993.
- [5] M.H. Halstead, “An Experiment Comparing Fortran Programming Times With The Software Physics Hypothesis,” Software Engineering Metrics I: Measure and Validations, Vol. 1, pp. 17-21, 1993.
- [6] E.I. Oviedo, “Control Flow, Data Flow And Program Complexity,” Software Engineering Metrics I: Measure and Validations, Vol. 1, pp. 52-65, 1993.
- [7] สยามคอมพิวเตอร์, “พัฒนาโปรแกรมด้วย Visual Basic,” กรุงเทพมหานคร: ไทยเจริญการพิมพ์, 2538.
- [8] T. Gilb, “Software Metrics,” Cambridge:Winthrop Publisher, Inc., 1977.
- [9] N.E. Fenton and Shari Lawrence Pfleeger, “Software Metrics: A Rigorous and Practical Approach,” 2 ed, FWS Publishing Company, 1997.
- [10] R.S. Pressman, “Software Engineering : A practitioner’s approach,” 4 ed, McGraw – Hill, 1997.
- [11] นิภาพร ประภาศิริ, “การวัดความซับซ้อนของซอฟต์แวร์,” วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ, สจล, 2544.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

