บทที่ 1

บทนำ

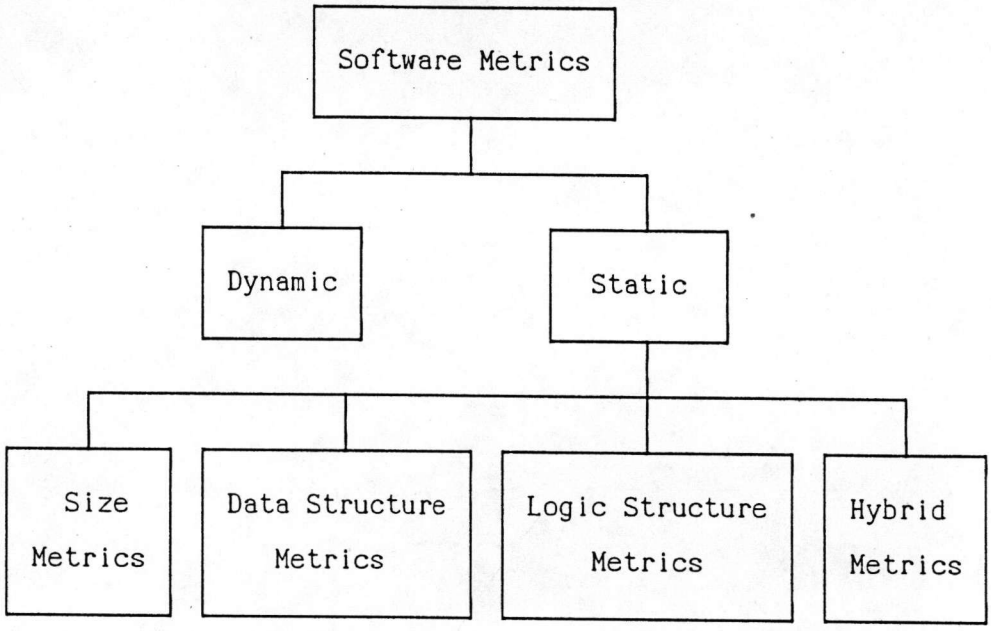
1.1 ความเป็นมาของปัญหา

การพัฒนาระบบงานคอมพิวเตอร์เป็นกิจกรรมที่ยุ่งยากและซับซ้อน ต้องใช้ทั้งเวลาและผู้พัฒนาระบบเป็นจำนวนมาก ความสำเร็จของโครงงานเช่นนี้จะต้องมีองค์ประกอบด้านการบริหารและเทคนิคที่ดี การควบคุมคุณภาพและค่าใช้จ่ายในการพัฒนาโปรแกรมนับเป็นกระบวนการที่มีความสำคัญมาก ซึ่งถ้าหากโปรแกรมที่เขียนขึ้นขาดคุณสมบัติที่ดี เช่น มีลักษณะโครงสร้างของโปรแกรมที่ไม่ดี มีความซับซ้อนมากอาจจะต้องใช้เวลาเขียนนาน เขียนแล้วอาจมีจุดผิดพลาดมาก นอกจากนี้ยังทำให้การซ่อมบำรุง การทดสอบโปรแกรม รวมทั้งการแก้ไขเปลี่ยนแปลงทำได้ยากมากขึ้นอีกด้วย

ในช่วงเวลา 20 ปีที่ผ่านมานี้ได้เริ่มมีการคิดค้นวิธีการวัดความซับซ้อนของซอฟต์แวร์ที่เรียกว่า ตัววัดซอฟต์แวร์ (Software Metrics) ซึ่งเป็นแนวความคิดใหม่ สามารถนำมาประยุกต์สร้างเป็นเครื่องมือในการติดตามผลงานของโครงงาน และวัดคุณภาพของโปรแกรม การหาค่าความซับซ้อนอาจประมาณง่ายๆจาก ขนาดของโปรแกรม จำนวนของคำสั่ง ขนาดของหน่วยความจำที่ต้องใช้ จำนวนของตัวดำเนินการ (operator) และตัวถูกดำเนินการ (operand) ความยากง่ายของแบบการคำนวณที่ใช้ เป็นต้น ซึ่งต้องยอมรับว่า เริ่มแรกสาขานี้ยังไม่มีพื้นฐานทางทฤษฎีที่ดีนัก ในอดีตงานวิจัยเหล่านี้จึงไม่ได้นำมาประยุกต์เพื่อใช้งานกันอย่างจริงจัง แต่ต่อมาก็เป็นสิ่งที่ยอมรับได้ว่ามีประโยชน์สามารถนำไปใช้กับงานจริงได้ ขณะนี้บริษัทคอมพิวเตอร์ชั้นนำหลายแห่งได้พยายามนำตัววัดมาสร้างเป็นเครื่องมือมาตรฐาน เพื่อใช้ในบริษัท แต่ส่วนใหญ่แล้วก็กำลังอยู่ในขั้นตอนของการพัฒนา

ความหมายของคำว่าตัววัดซอฟต์แวร์คือ วิธีมาตรฐานที่ใช้ในการวัดคุณสมบัติบางอย่าง เช่น ขนาด ค่าใช้จ่าย จุดบกพร่อง ความยากง่าย และสภาพแวดล้อมต่างๆของโปรแกรม (Grady and Caswell, 1987) ส่วนความหมายของคำว่าความซับซ้อนของซอฟต์แวร์คือความยากต่อการเข้าใจ การแก้ไขเปลี่ยนแปลง และการทดสอบโปรแกรม (เอื้อน ปิ่นเงิน, 2535)

การวัดความซับซ้อนของซอฟต์แวร์ที่เคยมีการพัฒนาขึ้นมา อาจแบ่งเป็นประเภทต่างๆดังแสดงในรูปที่ 1.1 คือแบ่งเป็นรูปแบบพื้นฐาน 2 แบบคือ สแตติก และไดนามิก (Waramahatuti et al., 1988) ตัววัดแบบสแตติกจะทำการวัดค่าความซับซ้อนโดยวิเคราะห์จากตัวรหัส (code) ของโปรแกรม ส่วนแบบไดนามิกจะวัดในช่วงเวลาที่โปรแกรมนั้นทำงานอยู่ สำหรับงานวิจัยนี้จะทำการศึกษาเฉพาะแบบสแตติกเท่านั้น



รูปที่ 1.1 แสดงการแบ่งประเภทของตัววัดซอฟต์แวร์

การวัดแบบสแตติกแบ่งลักษณะการวัดได้อีก 4 แบบ (Conte et al., 1987) ดังนี้

ก) ตัววัดขนาด (Size Metrics) เป็นทฤษฎีการวัดที่ถูกสร้างขึ้นเป็นตัวแรก สามารถทำความเข้าใจได้ง่าย และนิยมใช้มากที่สุด ส่วนใหญ่เป็นการประมาณขนาดของโปรแกรมโดยอาศัยการนับคำแล้วนำมาคำนวณหาค่าความซับซ้อน ตัววัดที่จัดอยู่ในประเภทนี้ได้แก่ จำนวนบรรทัดของโปรแกรม (Lines of code) ซึ่งหาค่าได้จากนับจำนวนบรรทัดของโปรแกรม และตัววัดฮอลสตีด (Halstead's Metric) บางรูปแบบ เช่น ปริมาตร ความยาวของโปรแกรม

ข) ตัววัดโครงสร้างข้อมูล (Data Structure Metrics) วิธีนี้จะคำนวณค่าความซับซ้อนโดยพิจารณาจาก ปริมาณ โครงสร้าง การสื่อสาร และการใช้ข้อมูลร่วมกันระหว่างโมดูล การจัดการและการกระจายของข้อมูล ค่าคงที่ และตัวแปรต่างๆ ภายในโปรแกรม

ค) ตัววัดโครงสร้างตรรกะ (Logic Structure Metrics) เป็นการหาความซับซ้อนจากโปรแกรมกราฟ หรือที่เรียกได้อีกชื่อหนึ่งว่า กราฟกระแสควบคุม (control flow graph : CFG) ตัวอย่างของตัววัดแบบนี้ ได้แก่ ตัววัดเชน ตัววัดแมคคลู ตัววัดไซโคลแมติกของแมคเคบ

ง) ตัววัดแบบผสม (Hybrid Metrics) เป็นตัววัดที่หาค่าความซับซ้อนจากกระแสข้อมูล (data flow) และจากกระแสการควบคุม (control flow) ตัววัดที่จัดอยู่ในประเภทนี้ได้แก่ ตัววัดโอวิโด (Oviedo's Metrics)

นักวิชาการที่สร้างทฤษฎีหาค่าความซับซ้อนแต่และแบบต่างก็มีมุมมองที่ต่างกัน และตัววัดแต่ละแบบก็เหมาะกับลักษณะงานที่ต่างกัน ดังนั้นการใช้ตัววัดเพียงตัวเดียวในระบบงานหนึ่งๆ อาจไม่เหมาะสม เพราะไม่สามารถระบุถึงสิ่งที่ต้องการทราบและต้องการทำการควบคุมได้ทั้งหมด

งานวิจัยนี้จะทำการรวบรวมตัววัดซอฟต์แวร์หลายๆแบบ แล้วนำมาออกแบบระบบสร้างเป็นโปรแกรมที่สามารถคำนวณค่าความซับซ้อนแบบต่างๆของภาษาซี โดยมีจุดประสงค์เพื่อให้สามารถวัดความซับซ้อนของโปรแกรม เพื่อใช้เป็นเกณฑ์ในการติดตามผลงาน คุณภาพและบ่งบอกถึงระบบหรือส่วนของโปรแกรมบางส่วนที่อาจมีปัญหา

1.2 วัตถุประสงค์ของการวิจัย

เพื่อศึกษาและพัฒนา โปรแกรมการวัดความซับซ้อนของซอฟต์แวร์มาประยุกต์ใช้งานโดยอาศัยหลักทฤษฎีของตัววัดซอฟต์แวร์ (software metrics)

1.3 ขั้นตอนการวิจัย

1.3.1 ศึกษาทฤษฎีของตัววัดความซับซ้อนของซอฟต์แวร์แบบต่างๆ
1.3.2 วิเคราะห์และเลือกเทคนิคที่จะนำมาใช้สร้างระบบวัดความซับซ้อนของซอฟต์แวร์
1.3.3 ออกแบบระบบ
1.3.4 พัฒนาโปรแกรม
1.3.5 ทดสอบประสิทธิภาพและความถูกต้องของระบบ
1.3.6 สรุปผลการวิจัย และข้อเสนอแนะ

1.4 ขอบเขตการวิจัย

1.4.1 ในการวิจัยครั้งนี้จะเลือกศึกษาตัววัดซอฟต์แวร์บางแบบเท่านั้น โดยจะวัดค่า

1.4.1.1 ตัววัดของฮอลสตีด (Halstead's Metrics)

1.4.1.1.1 ความยาวโปรแกรม

1.4.1.1.2 ปริมาตร

1.4.1.1.3 ปริมาตรศักยะ

1.4.1.1.4 ระดับของการโปรแกรม

1.4.1.1.5 ระดับภาษา

1.4.1.1.6 ความล้ำลึกของขั้นตอนกรรมวิธี

1.4.1.2 ตัววัดโครงสร้างตรรกะ (Logic Structure Metrics) โดยวิธี

1.4.1.2.1 ตัววัดเชน (Chen Metrics)

1.4.1.2.2 ตัววัดแมคคลู (McClure Metrics)

1.4.1.3 ตัววัดแบบผสม (Hybrid Metrics) โดยใช้วิธีตัววัดของโอวิโด (Oviedo's Metrics)

1.4.2 พัฒนาโปรแกรมบนเครื่องคอมพิวเตอร์ระดับ ไมโครคอมพิวเตอร์

1.4.3 ใช้ภาษา ซี ในการเขียนโปรแกรมวัดความซับซ้อนของซอฟต์แวร์

1.4.4 โปรแกรมที่จะนำมาวัดความซับซ้อนได้เป็นโปรแกรมที่เขียนด้วยภาษา ซี เท่านั้น

1.4.5 โปรแกรมที่นำมาวัดต้องเป็นโปรแกรมที่มีความถูกต้องทางไวยากรณ์ของภาษาซีแล้ว

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้โปรแกรมที่มีความสามารถวัดความซับซ้อนของโปรแกรมภาษาซี

1.5.2 เพื่อใช้วัดประสิทธิภาพของโปรแกรม นำมาใช้เป็นแนวทางในการออกแบบและการควบคุมคุณภาพของระบบให้เป็นตามที่ต้องการได้

1.5.3 เพื่อใช้เป็นบรรทัดฐานในการติดตามแก้ไขโปรแกรม

1.5.4 ใช้เป็นแนวทางเพื่อบ่งบอกถึงส่วนของโปรแกรมหรือโมดูลที่อาจจะมีปัญหา

1.5.5 ใช้เป็นแนวทางในการควบคุมความก้าวหน้าของโครงงาน และใช้เป็นข้อมูลเพื่อพิจารณางบประมาณสำหรับระยะต่างๆ ของโครงงาน

1.5.6 ช่วยเพิ่มประสิทธิภาพ ลดค่าใช้จ่ายในการพัฒนาและการบำรุงรักษาระบบ