บทที่ 6

บทสรุป และ ข้อเสนอแนะ

6.1 สรุปการวิจัย

จากการพัฒนาโปรแกรมการวัดความซับซ้อน โดยอาศัยหลักทฤษฎีของตัววัดซอฟต์แวร์ (software metrics) ผลการวิจัยปรากฏว่า โปรแกรมตัววัดที่สร้างจากทฤษฎีการวัดทั้งสี่แบบนี้ เมื่อนำค่าที่วัดได้ทั้งหมดมาประกอบกันโดยรวมๆแล้วสามารถใช้เป็นเกณฑ์พิจารณาประสิทธิภาพของโปรแกรมได้ดี แต่อาจต้องดูองค์ประกอบอื่นๆด้วย เช่น ประเภทของการใช้งานว่าต้องการความเร็วการทำงานสูงหรือต้องการให้โปรแกรมมีขนาดเล็ก เพราะโปรแกรมที่มีขนาดเล็กเมื่อวัดด้วยตัววัดเหล่านี้ บางครั้งจะมีความซับซ้อนน้อยกว่าแต่ทำงานได้ช้ากว่าโปรแกรมที่มีขนาดใหญ่กว่า และสำหรับตัววัดเชนจะมีจุดบกพร่องอยู่ คืออาจวัดค่าได้ไม่เที่ยงตรงนักในโปรแกรมบางลักษณะเช่น การซ้อนใน (nesting) ผู้วิจัยได้เปรียบเทียบค่าที่วัดได้กับตัววัดไซโคลแมติกหรือตัววัดแมคคลูซึ่งเป็นตัววัดโครงสร้างตรรกะเช่นเดียวกันแล้ว พบว่าตัววัดไซโคลแมติกและตัววัดแมคคลูได้ค่าที่น่าเชื่อถือกว่า อนึ่งการวัดความซับซ้อนนี้จะไม่สามารถรวมค่าของความซับซ้อนของในส่วนของฟังก์ชันจากไลบรารีมาตรฐาน (standard library) ต่างๆที่ถูกเรียกจากโปรแกรมที่นำมาวัด

6.2 ข้อเสนอแนะ

เนื่องจากค่าความซับซ้อนนี้เป็นเพียงตัวเลข ซึ่งไม่มีค่ามาตรฐานที่แน่นอนในการเปรียบเทียบ ยกเว้นตัววัดบางแบบเช่น แมคเคบกำหนดว่าแต่ละโมดูลไม่ควรมีค่า m เกิน 10 (เอื้อน บิ่นเงิน, 2535) จึงยากที่จะกล่าวได้ว่าโปรแกรมที่ทำการวัดนี้มีประสิทธิภาพอย่างไร นอกจากจะทำการเปรียบเทียบกันเองระหว่างโปรแกรมที่ทำงานในลักษณะเดียวกัน ดังนั้นบางครั้งค่าความซับซ้อนอาจใช้ประโยชน์ได้ในแง่ที่เป็นค่าซึ่งบอกถึงข้อมูลบางประการที่ได้จากโปรแกรมที่นำมาวัด และจะใช้ประโยชน์ได้มากขึ้น ถ้าหากมีการกำหนดค่ามาตรฐานของตัววัดแต่ละแบบ

6.3 แนวทางการวิจัยต่อ

6.3.1 พัฒนาโปรแกรมวัดความซับซ้อนจากทฤษฎีของนักวิจัยท่านอื่นๆ

6.3.2 สร้างทฤษฎีการวัดความซับซ้อนขึ้นใหม่ แล้วทดสอบว่าตัววัดแบบใหม่นี้สามารถทำนายค่าความซับซ้อนได้ดีเพียงใด

6.3.3 พัฒนาโปรแกรมนี้ให้สามารถวัดค่าความซับซ้อนจากโปรแกรมที่เขียนด้วยภาษาอื่นๆ เช่น ปาสคาล โคบอล ฟอร์แทรน