

งานวิจัยนี้มุ่งหวังที่จะศึกษาลักษณะของโปรแกรมตัวอย่าง เพื่อวิเคราะห์ความสัมพันธ์ของตัววัดซอฟต์แวร์ชนิดต่าง ๆ โดยนำตัววัดซอฟต์แวร์ (Software Metric) ชนิดต่าง ๆ มาหาความสัมพันธ์แบบที่ไม่ نرمัลไลซ์ (Non-Normalize) และแบบที่ نرمัลไลซ์ (Normalize) โดยโปรแกรมตัวอย่างที่นำมาใช้ในงานวิจัยนี้เป็นโปรแกรมขนาดเล็กที่เขียนด้วยภาษาจาวา มีจำนวน 2 กลุ่มโปรแกรมตัวอย่าง คือกลุ่มโปรแกรมตัวอย่างที่ 1 จำนวน 36 โปรแกรม มีจำนวนบรรทัดของโปรแกรมใกล้เคียงกันและกลุ่มโปรแกรมตัวอย่างที่ 2 จำนวน 37 โปรแกรม มีจำนวนบรรทัดของโปรแกรมแตกต่างกัน ตัววัดซอฟต์แวร์ที่นำมาใช้มีทั้งหมด 7 ตัว คือความซับซ้อนไซโคลมาติก (Cyclomatic Complexity) ดับเบิลยูเอ็มซี (Weighted Methods per Class: WMC) อาร์เอฟซี (Response for a Class: RFC) ซีบีโอ (Coupling between Object Classes: CBO) แอลซีโอเอ็ม (Lack of Cohesion of Method: LCOM) แฟน-อิน (Fan-in) และแฟน-เอาท์ (Fan-out) โดยใช้สถิติในการหาความสัมพันธ์แบบสหสัมพันธ์ของเพียร์สัน (Pearson's Correlation) หลังจากนั้นทำการเปรียบเทียบความสัมพันธ์ของตัววัดซอฟต์แวร์ชนิดต่าง ๆ ของโอเพ่นซอร์สซอฟต์แวร์กับโปรแกรมโครงงานของนักศึกษาชั้นปีที่ 4 คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี พบว่านักศึกษายังมีจุดด้อยในการพัฒนาซอฟต์แวร์อยู่หลายด้าน เช่นการแบ่งจำนวนเมทอด จำนวนคลาส และความซับซ้อนของโปรแกรม

This thesis studies the complexity of Java Programs using selected software metrics to measure the complexity in order to plan software tests and maintenance. The sample programs used in this thesis small-size Java Programs which were Open Source Software. There were two sample groups: the first one contained thirty-six programs. The second group had thirty-seven programs. The study employed the following seven software metrics to measure the programs' complexity: Cyclomatic Complexity, Weighted Method per Class (WMC), Response for a Class (RFC), Coupling between Object Classes (CBO), Lack of Cohesion of Method (LCOM), Fan-in, and Fan-out. Pearson's correlation method was used to find the correlation between the number of Line of Code (LOC) and each of the seven software metrics, and the correlations among those seven software metrics. All correlations were determined both at the program level (not normalized) and at the class level (normalized). The study also compared the complexity of Open Source Software to the programs written by the fourth year students of the school of Information Technology at King Mongkut's University of Technology Thonburi. Some weak points in the students' software were identified.