



โมเดลประเมินคุณภาพด้านความสามารถในการวิเคราะห์เพื่อหาข้อผิดพลาด สำหรับโค้ดจาวาระดับกลาง

Quality Model for Analyzability Evaluation Of Java Class

จุฑามาส สงวนจิตร^{1*} ทรงศักดิ์ ร่องวิริยะพานิช¹ และ พนิดา เมณะเนตร¹

บทคัดย่อ

งานวิจัยนี้เสนอโมเดลประเมินคุณภาพของซอร์สโค้ดด้านความสามารถในการวิเคราะห์เพื่อหาข้อผิดพลาด (Analyzability) สำหรับโค้ดจาวาระดับกลาง ซึ่งจะช่วยให้ระดับความยากง่ายในการวิเคราะห์หาสาเหตุของข้อผิดพลาดและตำแหน่งที่ต้องแก้ไข ในการสร้างโมเดลผู้วิจัยได้สำรวจหามาตรวัดที่คาดว่าจะมีผลต่อคุณภาพของซอร์สโค้ดด้านการวิเคราะห์หาสาเหตุข้อผิดพลาดและตำแหน่งที่ต้องแก้ไข และใช้วิธีทางสถิติการถดถอยโลจิสติกแบบอันดับ เพื่อค้นหามาตรวัดที่มีความสัมพันธ์กับระดับคุณภาพของซอร์สโค้ดด้านการวิเคราะห์หาสาเหตุข้อผิดพลาดและตำแหน่งที่ต้องแก้ไข ซึ่งแบ่งไว้ 3 ระดับ คือ 1. ควรปรับปรุง 2. พอใช้ 3. ดี หลังจากนั้นได้ทำการทดสอบวัดความถูกต้องแม่นยำของโมเดลที่ระดับความเชื่อมั่น 95% โดยทดสอบกับกลุ่มตัวอย่างคลาสจาวาจากซอฟต์แวร์โอเพ่นซอร์ส jEdit จำนวน 37 คลาสที่มีการบันทึกว่ามีข้อผิดพลาดและให้ผู้เชี่ยวชาญได้เข้าไปวิเคราะห์หาจุดที่จะต้องแก้ไขและให้ค่าระดับคุณภาพของคลาส จากการทดลองพบว่ามาตรวัดที่มีผลต่อคุณภาพของซอร์สโค้ดด้านความสามารถในการวิเคราะห์เพื่อหาข้อผิดพลาด คือ Coupling Between Object (CBO) และ Lack Of Cohesion In Methods (LCOM)

¹สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

*Corresponding Author: jutamas@nstda.or.th

ABSTRACT

We propose a quality model for evaluating the analyzability of java class. Our model determines how difficult to understand the code in order to identify cause and error locations. To create this prediction model, we survey software metrics likely to have an impact on class's analyzability. We applied the ordinal logistic regression method to identify a set of metrics correlated with the analyzability level divided into 3 levels: poor, fair and excellent. To validate our model, we tested it at 95% confidence level using a set of 37 java classes from jEdit open source program. We found that only Coupling between Object (CBO) and Lack of Cohesion in Methods (LCOM) metrics influence class analyzability.

คำสำคัญ: โมเดลวัดคุณภาพซอฟต์แวร์ ความสามารถในการบำรุงรักษาซอฟต์แวร์ โมเดลวัดคุณภาพซอร์สโค้ด ด้านความสามารถในการวิเคราะห์เพื่อหาข้อผิดพลาด การถดถอยโลจิสติกแบบอันดับ มาตรฐาน

Keywords: Software Quality Model, Software Maintainability, Analyzability Model, Ordinal Logistic Regression, Metric

บทนำ

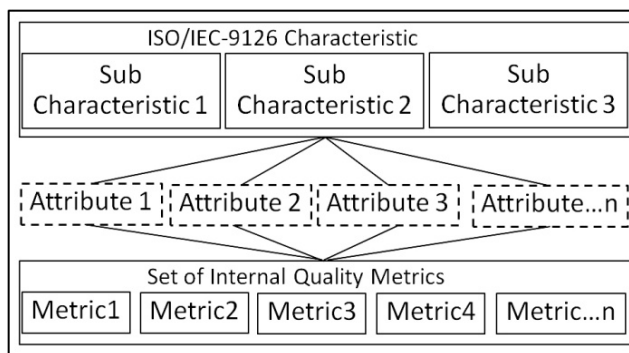
การออกแบบซอฟต์แวร์รวมทั้งการเขียนโค้ดให้สามารถอ่านทำความเข้าใจง่ายเป็นสิ่งจำเป็นสำหรับการบำรุงรักษาซอฟต์แวร์ เนื่องจากในการบำรุงรักษาโค้ดทั้งแบบการแก้ไขข้อผิดพลาดและแบบเพิ่มเติมคุณสมบัติ จำเป็นที่โค้ดต้องได้รับการออกแบบให้สามารถหาสาเหตุข้อผิดพลาดและจุดแก้ไขได้ง่าย เนื่องจากจะช่วยลดเวลาในการหาตำแหน่งที่นักพัฒนาโปรแกรมต้องแก้ไขเพื่อปรับปรุงหรือพัฒนาต่อยอด โค้ดโปรแกรมที่มีคุณสมบัติง่ายต่อการทำความเข้าใจที่ให้นักพัฒนาสามารถวิเคราะห์หาสาเหตุข้อผิดพลาดและแก้ไขโปรแกรมได้ง่ายนั้นในเชิงเทคนิคเรียกว่า โค้ดมีความสามารถในการวิเคราะห์เพื่อหาข้อผิดพลาด (Analyzability) สูง ซึ่งถือว่าเป็นคุณสมบัติหนึ่งที่ส่งเสริมให้ซอฟต์แวร์มีคุณสมบัติด้านการบำรุงรักษาที่ง่าย (Maintainability) ตามไปด้วย จากเหตุผลดังกล่าวทำให้เกิดแนวคิดที่ต้องการหาโมเดลหรือวิธีการพยากรณ์คุณภาพของซอร์สโค้ดด้านความสามารถในการวิเคราะห์เพื่อหาข้อผิดพลาด (Analyzability Model) ในงานวิจัยนี้มุ่งเน้นไปที่การสร้างโมเดลพยากรณ์คุณภาพด้าน Analyzability ระดับคลาส เนื่องจากการวัดคุณภาพของซอร์สโค้ดระดับคลาสทำให้สามารถคำนวณคุณภาพของซอร์สโค้ดทั้งระบบได้ (Kanellopoulos et al, 2010) และสามารถเป็นแนวทางในการสร้างโมเดล Analyzability วัดที่ระดับอื่น หรือวัดคุณสมบัติด้านอื่นๆของซอฟต์แวร์ได้

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1 ทฤษฎีที่เกี่ยวข้อง

1.1 ISO/IEC-9126 เป็นมาตรฐานวัดคุณภาพซอฟต์แวร์สามารถวัดคุณภาพจากซอร์สโค้ด (Internal Quality) โดยคุณลักษณะซอฟต์แวร์ (Characteristic) สามารถแยกเป็นคุณลักษณะย่อยได้ (Sub Characteristic)

คุณลักษณะย่อยแต่ละด้านจะประกอบขึ้นจากกลุ่มของแอททริบิวต์ และแอททริบิวต์จะสามารถวัดได้จากค่ามาตรวัด (Metric) โดยคุณลักษณะของซอฟต์แวร์ คุณลักษณะย่อย แอททริบิวต์ และมาตรวัดมีความสัมพันธ์เชิงลำดับชั้น ดังแสดงในรูปที่ 1 วิธีการประเมินคุณภาพของซอร์สโค้ดแบบ Analytic Hierarchy Process (AHP) (Kanellouopoulos et al., 2010) ได้ระบุถึงการใช้อนุกรมวัดเพื่อประเมินค่าแอททริบิวต์ คุณลักษณะย่อย ตามลำดับชั้นจนได้ค่าคุณภาพของซอร์สโค้ดในที่สุด



รูปที่ 1 โครงสร้างของคุณลักษณะเชิงคุณภาพ ISO/IEC-9126

ตัวอย่างเช่น ISO/IEC-9126 ได้นิยามคุณภาพด้านการบำรุงรักษา (Maintainability) คือ ความสามารถของซอฟต์แวร์ที่สามารถปรับเปลี่ยนแก้ไขให้ถูกต้องและสมบูรณ์ เมื่อจำเป็นต้องปรับเปลี่ยนความต้องการเดิมหรือมีความต้องการใหม่ ซึ่งประกอบจากคุณลักษณะย่อย 5 ประการ (Antonellis et al., 2007) คือ

- 1) ความสามารถด้านการวิเคราะห์ข้อผิดพลาด (Analyzability) คือ ระดับความยากง่ายในการวิเคราะห์หาข้อบกพร่อง สาเหตุของความล้มเหลว หรือ จุดที่ต้องแก้ไข
- 2) ความสามารถด้านการปรับเปลี่ยน (Changeability) คือ ระดับความยากง่ายในการปรับเปลี่ยน แก้ไข
- 3) ระดับความเสถียรของซอฟต์แวร์หลังแก้ไข (Stability) คือ ระดับความเสถียรของการทำงานของซอฟต์แวร์ภายหลังการแก้ไข
- 4) ระดับความยากง่ายในการทดสอบซอฟต์แวร์ (Testability)
- 5) ระดับความสอดคล้องของซอร์สโค้ดกับมาตรฐานหรือธรรมเนียมปฏิบัติด้านการบำรุงรักษา (Maintainability Compliance)

และคุณลักษณะย่อยความสามารถด้านการวิเคราะห์ข้อผิดพลาด (Analyzability) ประกอบด้วยแอททริบิวต์ Volume Inheritance Complexity Coupling และ Cohesion แอททริบิวต์ Volume สามารถวัดจากมาตรวัด Class Lines of Code (CLOC) และ Data Access Metric (DAM) เป็นต้น

1.2 การถดถอยโลจิสติก (Logistic Regression Analysis) เป็นวิธีทางสถิติเพื่อศึกษาความสัมพันธ์ระหว่างตัวแปรอิสระและตัวแปรตาม และนำสมการที่ได้ไปพยากรณ์ตัวแปรตาม สามารถแบ่งได้เป็น 2 ประเภทตามค่าตัวแปรตาม (มะแอน, 2549; กัลยา, 2549) คือ

- 1) การถดถอยโลจิสติกแบบทวิ (Binomial Logistic Regression) จะใช้เมื่อตัวแปรตาม (Y) เป็นตัวแปรเชิงกลุ่มแบบนามบัญญัติ (Nominal Scale) คือ มีค่าได้เพียง 2 ค่า ได้แก่ 0 และ 1 เท่านั้น

2) การถดถอยโลจิสติกแบบพหุกลุ่ม (Multinomial Logistic Regression) จะใช้เมื่อตัวแปรตาม (Y) เป็นตัวแปรเชิงกลุ่มมีค่ามากกว่า 2 ค่า มี 2 แบบ คือ แบบนามบัญญัติ (Nominal Scale) และ แบบอันดับ (Ordinal Scale) ความแตกต่างของนามบัญญัติ (Nominal Scale) และ แบบอันดับ (Ordinal Scale) คือ แบบอันดับสามารถบอกอันดับและความแตกต่างได้แต่ไม่สามารถบอกระยะห่างของอันดับที่แน่นอนได้

2 งานวิจัยที่เกี่ยวข้อง

งานวิจัย (Rizvi et al., 2010; Khan et al., 2012; Panita et al., 2013) สร้างโมเดลวัดคุณภาพซอฟต์แวร์โดยทั้ง 3 งานวิจัยใช้วิธี Regression ในงานวิจัย (Rizvi et al., 2010) ใช้วิธีทางสถิติ Regression Analysis สร้างโมเดลพยากรณ์ความสามารถในด้านการบำรุงรักษาในระดับคลาส ชื่อ MEMOOD ซึ่งค่าความสามารถในการบำรุงรักษาคำนวณจากระดับความเข้าใจซอร์สโค้ด (Understandability) และ ความง่ายต่อการปรับเปลี่ยนแก้ไข (modifiability) โดยใช้มาตรวัดที่มีผลกับขนาด (Size) และความซับซ้อน (Structural Complexity) และคำนวณจากแผนภาพคลาส ข้อสังเกตงานวิจัยนี้วัดคุณลักษณะย่อยที่ไม่สอดคล้องกับมาตรฐาน ISO/IEC9126 และไม่ใช่แอททริบิวต์ในการวัดคุณภาพ

งานวิจัย (Khan et al., 2012) สร้างโมเดลวัดคุณภาพด้านความสามารถด้านการวิเคราะห์ข้อผิดพลาด (Analyzability) ด้วยวิธีทางสถิติ Multiple Linear Regression ผลจากงานวิจัยพบว่าการออกแบบที่ซับซ้อน (Complexity) เป็นปัจจัยหลักที่ส่งผลต่อคุณภาพด้านความสามารถด้านการวิเคราะห์ข้อผิดพลาด (Analyzability) จึงได้นำมาตรวัดที่เกี่ยวข้องกับความซับซ้อนมาใช้เป็นหลัก มีข้อสังเกตว่าในงานวิจัยอาจมีปัจจัยอื่นที่เกี่ยวข้องที่ทำให้การวิเคราะห์ข้อผิดพลาดทำได้ยากขึ้น เช่น ขนาดของโปรแกรม

งานวิจัย (Panita et al., 2013) สร้างโมเดลวัดคุณภาพด้านความสามารถด้านการวิเคราะห์ข้อผิดพลาด (Analyzability) ที่ระดับเมธอดสำหรับ J2EE Application โดยโมเดลได้ประยุกต์ใช้โครงสร้างความสัมพันธ์ของ SIG Quality Model (Heitlager et al., 2007) และใช้วิธีการทางสถิติ Ordinal Logistic Regression

งานวิจัย (Antonellis et al., 2007; Kanellopoulos et al., 2010) สร้างโมเดลวัดคุณภาพซอฟต์แวร์ด้วยวิธีการที่ไม่ใช่วิธี Regression งานวิจัย (Kanellopoulos et al., 2010) เสนอวิธีการประเมินซอร์สโค้ดตามโครงสร้าง ISO/IEC-9126 และเสนอเพิ่มแอททริบิวต์ที่มีผลต่อคุณภาพตาม ISO/IEC-9126 มีการให้น้ำหนักมาตรวัดด้วยวิธี Analytic Hierarchy Process (AHP) ใช้วิธีการตัดสินใจแบบ Multi-Criteria ที่มาจากผู้เชี่ยวชาญ มีข้อสังเกตว่างานวิจัยนี้ยังขาดการทดสอบความถูกต้องของโมเดลที่เสนอ

งานวิจัย (Antonellis et al., 2007) เสนอวิธีประเมินคุณภาพด้านการบำรุงรักษา (Maintainability) ตามหลักการ ISO/IEC-9126 ด้วยวิธี Data Mining โดยใช้เทคนิค Analytical Hierarchy Process (AHP) และใช้เทคนิค K-Means ในการทำ Clustering ช่วยคัดกรองข้อมูลที่มีลักษณะใกล้เคียงกันเข้ากลุ่ม และวัด Error Sum of Squares เพื่อให้ Clustering ได้ผลดีขึ้น มีข้อสังเกตว่าในงานวิจัยไม่ใช่แอททริบิวต์สำหรับวัดคุณภาพ Maintainability ตามมาตรฐาน ISO/IEC-9126 และสูตรการคำนวณค่ามาตรวัดไม่ได้ระบุไว้ชัดเจน บอกเพียงเป็นมาตรวัด Metric Suite ของ Chidamber & Kemerer (Chidamber et al., 1994) ซึ่งจำนวนมาตรวัดที่ใช้ไม่สอดคล้องกับที่เสนอในงานวิจัยของ Chidamber & Kemerer

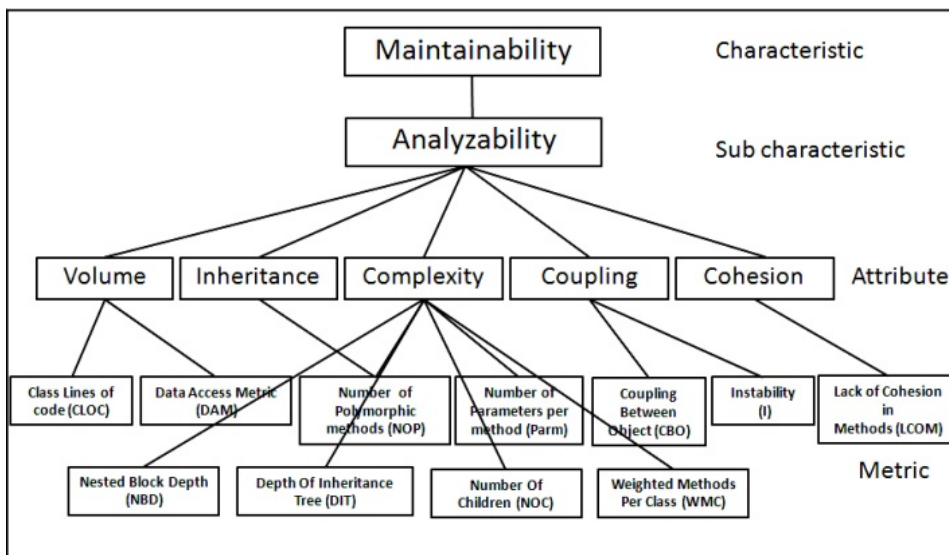
งานวิจัย (Lincke et al., 2008) มีข้อสังเกตเรื่องการนำเครื่องมือมาใช้เก็บค่ามาตรวัด เนื่องจากเครื่องมือแต่ละตัวจะให้ค่าต่างกันแม้จะวัดมาตรวัดเดียวกัน ดังนั้นในการสรุปผลวิจัยในการสร้างโมเดลพยากรณ์ใดที่คำนวณจากการใช้มาตรวัด ผู้วิจัยจะต้องระบุเครื่องมือที่ใช้ในการวัดค่ามาตรวัดด้วย

วิธีดำเนินการวิจัย

งานวิจัยนี้สร้างโมเดลโดยใช้วิธีการ Ordinal Logistic Regression มีการคัดเลือกมาตรวัดที่สามารถประเมินค่าครอบคลุมแอททริบิวต์ทั้ง 5 คือ Volume, Inheritance, Complexity, Coupling และ Cohesion ตามโครงสร้างลำดับชั้น ISO/IEC 9126 ดังรูปที่ 2 มาตรวัด (Metric) คัดเลือกจาก 2 แหล่ง คือ มาตรวัดจากงานวิจัยที่ผ่านมา (Kanellopoulos et al, 2010; Antonellis et al, 2007; Rudiger et al, 2008; Rizvi and Khan, 2010; Khan and Khan, 2012; Panita et al., 2013) และมาตรวัดจากการสัมภาษณ์ผู้เชี่ยวชาญ (ผู้มีประสบการณ์ด้านจาวามากกว่า 5 ปี) โดยวิธีการดำเนินงานวิจัยแบ่งเป็น 2 ขั้นตอน คือ 1 การกำหนดตัวแปรอิสระและตัวแปรตามของโมเดล Analyzability 2 การสร้างโมเดล Analyzability

1 การกำหนดตัวแปรอิสระและตัวแปรตามของโมเดล

ตัวแปรอิสระในโมเดล คือ มาตรวัดจำนวน 11 ตัว ตามตารางที่ 1 ที่สามารถประเมินค่าครอบคลุมแอททริบิวต์ทั้ง 5 ตามโครงสร้างระดับชั้น ISO/IEC 9126 และ ความสัมพันธ์ระหว่าง Analyzability กับแอททริบิวต์ และ มาตรวัดในงานวิจัยนี้แสดงดังรูปที่ 2



รูปที่ 2 โครงสร้าง analyzability

ตารางที่ 1 รายละเอียดมาตรวัดในงานวิจัย

มาตรวัด	จุดประสงค์	สูตรคำนวณ
Class Lines Of Code (CLOC)	วัดปริมาณซอร์สโค้ด	จำนวนบรรทัดคำสั่งในคลาส ไม่รวมบรรทัดว่างและคอมเม้นต์ (Kanellopoulos et al., 2010; Panita et al., 2013)
Data Access Metric (DAM)	วัดจำนวนแอททริบิวต์	อัตราส่วนจำนวนแอททริบิวต์ Private หรือ Protected ต่อจำนวนแอททริบิวต์ทั้งหมดที่ประกาศในคลาส (Antonellis et al., 2007)
Number Of Polymorphic Methods (NOP)	วัด Overridden หรือ Virtual เมธอด	จำนวนเมธอดที่ถูก Override โดยบรรพบุรุษในแต่ละคลาส (Antonellis et al., 2007)
Average Number Of Parameters Per Method (APARM)	วัดความซับซ้อนของพารามิเตอร์ต่อ 1 เมธอด	ค่าเฉลี่ยจำนวนพารามิเตอร์ต่อเมธอดของคลาส (Panita et al., 2013)
Coupling Between Object (CBO)	วัดจำนวนคลาสที่เกี่ยวข้อง (Couple) กับคลาสที่สนใจ ในลักษณะที่เป็น Use และ Member ไม่รวม Base Class และ Nested Class (Fan-Out)	จำนวนคลาสที่เกี่ยวข้อง (Couple) กับคลาสที่สนใจ ในลักษณะที่เป็น Use และ Member ไม่รวม Base Class และ Nested Class (Antonellis et al., 2007; Rudiger et al., 2008)
Instability (I)	วัด Coupling โดยเมื่อเปลี่ยนแปลงคลาสในแพ็คเกจแล้วจะไม่มีผลกระทบต่อคลาสในแพ็คเกจอื่น	$I = CE/(CA+CE)$ CE = Efferent Coupling คือ จำนวนคลาสในแพ็คเกจที่ขึ้นกับคลาสที่กำลังพิจารณา CA = Efferent Coupling คือ จำนวนคลาสนอกแพ็คเกจที่ขึ้นกับคลาสที่กำลังพิจารณา (Kanellopoulos et al., 2010; Panita et al., 2013)
Lack Of Cohesion In Methods (LCOM)	วัดการเกาะเกี่ยวกันในคลาส	$\frac{\left[\frac{1}{v} \sum_{i=1}^v m(v_i) \right] - m}{1 - m}$ m = จำนวนเมธอด v = จำนวนแอททริบิวต์ m(V _i) = เมธอดที่เรียกใช้แอททริบิวต์ V _i (Antonellis et al., 2007; Rudiger et al., 2008)
Nested Block Depth (NBD)	วัดความซับซ้อนของเงื่อนไขในคลาส	จำนวนระดับชั้นที่ลึกที่สุดของเงื่อนไข if while for switch ของคลาส (Antonellis et al., 2007; Rudiger et al., 2008)
Depth Of Inheritance (DI)	วัดความซับซ้อนจากการสืบทอดคุณสมบัติ	ความลึกของการสืบทอดคุณสมบัติของคลาส โดยนับระดับชั้นถัดไปของคลาส (Antonellis et al., 2007; Rudiger et al., 2008; Khan and Khan, 2012)
Number Of Children (NOC)	วัดความซับซ้อนของคลาสลูกที่สืบทอดจากคลาสที่สนใจ	จำนวนคลาสลูกที่สืบทอดจากคลาสที่สนใจ 1 ลำดับชั้น (Antonellis et al., 2007; Rudiger et al., 2008)
Weighted Methods Per Class (WMC)	วัดความซับซ้อนของเมธอดในคลาส	$WMC = \sum_{i=1}^n c_i$ ค่าเฉลี่ยของผลรวมของความซับซ้อนของเมธอดแต่ละเมธอดของคลาส(C) (Antonellis et al., 2007)

ตัวอย่างของคลาสที่นำมาใช้สร้างโมเดลประเมินคุณภาพด้านความสามารถในการวิเคราะห์โค้ดจาวาระดับคลาสมาจากโปรแกรม jEdit (jEdit Development Team, 2013) ที่สุ่มเลือกมาจำนวน 37 คลาสจากคลาสจำนวนมากในโปรแกรม jEdit สาเหตุที่ใช้โปรแกรมนี้เนื่องจากเป็นโปรแกรม Open Source และมีการบันทึกการแก้ไขข้อผิดพลาด (Bug Tracking) หรือความต้องการใหม่ ทำให้ผู้วิจัยสามารถตรวจสอบผลการวิเคราะห์และการประเมินคุณภาพโค้ดของผู้เชี่ยวชาญว่ามีความเข้าใจปัญหาที่แจ้งในรายการที่ให้แก้ไขจริงหรือไม่ และได้นำเครื่องมือ Eclipse Metric Plug-in 1.3.6 (Sourceforge) และ Understand 3.1.716 (Scientific Toolworks, Inc.) มาเก็บค่ามาตรวัดทั้ง 11 มาตรวัดของคลาส 37 คลาส เพื่อนำไปใช้เป็นค่าตัวแปรอิสระสำหรับสร้างโมเดล Analyzability

ตัวแปรตามในโมเดล คือ ความสามารถในการวิเคราะห์เพื่อหาข้อผิดพลาด (Analyzability) ค่าของตัวแปรได้จากการประเมินระดับคุณภาพของซอร์สโค้ดจาวา 37 คลาสโดยผู้เชี่ยวชาญ ด้วยระดับคุณภาพ 3 ระดับ ได้แก่ ระดับ 3 คือ คุณภาพดี 2 คือ พอใช้ 1 คือ แย่ ควรปรับปรุง รูปที่ 3 แสดงเกณฑ์ประเมินระดับ Analyzability ของผู้เชี่ยวชาญ เกณฑ์ที่ได้นี้ถูกนำไปใช้วิเคราะห์หามาตรวัดที่นำไปใช้สร้างโมเดล Analyzability ในลำดับถัดไปด้วย เช่นจำนวนบรรทัด ความซับซ้อนของเงื่อนไข หรือ ลูป

ลำดับ	เงื่อนไขการประเมินของผู้เชี่ยวชาญ	ระดับ Analyzability		
		1	2	3
1	ใช้เงื่อนไขซ้อนกันหลายชั้นหรือมีความซับซ้อน (เช่น if..else , while..loop,or,and)	>6	4-6	<4
2	จำนวนบรรทัดเยอะ	>2000	1000-2000	<1000
3	โล่งง่าย ข้อตัวแปร,เมธอดสื่อความหมาย,หาจากคีย์เวิร์คได้,จำนวนบรรทัดไม่มาก,แบ่งแยกการทำงานชัดเจน			/
4	fix ค่าตัวแปรในคลาส		/	
5	มี inner class ในคลาส		/	

รูปที่ 3 เกณฑ์ประเมินระดับ Analyzability ของผู้เชี่ยวชาญ

2 การสร้างโมเดล Analyzability

เนื่องจากตัวแปรตามของโมเดลเป็นระดับคุณภาพ 3 ระดับที่สามารถเรียงลำดับได้ ผู้วิจัยจึงใช้วิธีการ Ordinal Logistic Regression ในการสร้างโมเดล ซึ่งสมการเริ่มต้นสำหรับการสร้างโมเดลจะอยู่ในรูปสมการความสัมพันธ์ระหว่างตัวแปรตาม (Y) คือ ค่าระดับคุณภาพ Analyzability ที่ได้จากการประเมินโดยผู้เชี่ยวชาญ และตัวแปรอิสระ (X) คือ ค่ามาตรวัดจำนวน 11 ตัว ตามสมการ (1) รูปที่ 4 แสดงตัวอย่างค่ามาตรวัดและ Analyzability ของคลาสกลุ่มตัวอย่าง

ลำดับคลาส	CLOC	DAM	NOP	APARM	CBO	I	LCOM	NBD	DI	NOC	WMC	Analyzability
												ระดับคุณภาพ
1	563	0	0	3.091	4	0.163	0	6	1	0	7	2
2	361	0.8	1	1.652	2	0.023	0	4	1	0	3	2
3	235	0.75	6	1.043	5	0.511	0.222	3	2	3	2	3
4	288	0.556	5	1.235	10	0.163	0.475	3	1	0	4	1

รูปที่ 4 ค่ามาตรวัดและระดับคุณภาพด้าน Analyzability ของคลาสตัวอย่าง

โมเดลทางคณิตศาสตร์ของ Logistic Regression ตามสมการ (1)

$$g(x) = \text{Logit}[\pi(x)] = \log_e \left[\frac{\pi(x)}{1-\pi(x)} \right] = \alpha + \beta_1 x_1 + \dots + \beta_m x_m \quad (1)$$

$g(X)$ เรียกว่า Link Function คือ ข้อมูลของตัวแปรตามในแต่ละกลุ่มของระดับชั้นนั้น ๆ และ X_1, X_2, \dots, X_{11} คือ ตัวแปรอิสระ (X) เป็นตัวแปรต่อเนื่องมีขนาด m ในสมการ (1) = 11 ตัว

ความสัมพันธ์ระหว่างตัวแปรอิสระและตัวแปรตามที่มีค่าเป็นกลุ่มและเรียงอันดับได้ค่าตัวแปรตามพิจารณาได้จากความน่าจะเป็น (Probability) ของแต่ละค่าระดับ โดยเลือกระดับที่มีความน่าจะเป็นสูงสุด ทั้งนี้ความน่าจะเป็นของระดับจะถูกคำนวณความน่าจะเป็นสะสม (Cumulative Probability) ตามสมการ (2)

$$P(Y \leq i) = \frac{1}{1 + \exp\left(-(\alpha + \beta_1 X_1 + \dots + \beta_m X_m)\right)}, \quad i=1, \dots, k \quad (2)$$

เมื่อค่า k คือ จำนวนของค่าระดับของตัวแปรตามลบหนึ่ง ดังนั้น $k=2$ และ m คือ จำนวนตัวแปรอิสระ ดังนั้น $m=11$ ทั้งนี้ความน่าจะเป็นสะสมของระดับสุดท้ายจะมีค่าเป็น 1 เสมอ นั่นคือ $P(Y \leq 3) = 1$ และจำนวนของสมการ (2) เท่ากับ จำนวนของค่าระดับของตัวแปรตามลบหนึ่ง หรือ 2 นั่นเอง จากสมการ (2) สามารถหาค่าความน่าจะเป็นของแต่ละระดับได้ด้วยตามสมการ (3)

$$P(Y=i) = P(Y \leq i) - P(Y < i) \quad (3)$$

2.1 การตรวจสอบเงื่อนไขการใช้การถดถอยโลจิสติก

นำคลาส 37 คลาสมาตรวจสอบเงื่อนไขการใช้การถดถอยโลจิสติก (กัลยา, 2549) โดยใช้โปรแกรม SPSS (IBM) ที่ระดับความเชื่อมั่น 95% เนื่องจากมาตรวัดเป็นข้อมูลชนิด Ratio Scale และสมการถดถอยมีค่าเฉลี่ยของความคลาดเคลื่อนเท่ากับ 0 หรือไม่สัมพันธ์กัน ($e=0$) จึงทำการตรวจสอบใน 3 ข้อดังนี้

1) การตรวจสอบค่าความคลาดเคลื่อนเป็นอิสระต่อกันของ e_i และ e_j จากการตรวจสอบ Durbin-Watson เท่ากับ 1.974 แสดงว่าค่าความคลาดเคลื่อนของมาตรวัดเป็นอิสระต่อกัน

2) การตรวจสอบค่าความคลาดเคลื่อนไม่สัมพันธ์กับตัวแปรอิสระหรือไม่มีความสัมพันธ์กันสูง จากการตรวจสอบ Tolerance ของมาตรวัดแต่ละตัว พบว่าค่า Tolerance มากสุดอยู่ที่ 0.772 แสดงว่าค่าความคลาดเคลื่อนมาตรวัดไม่สัมพันธ์กัน

3) ตัวแปรอิสระไม่ควรมีความสัมพันธ์กันหรือไม่ควรเกิดปัญหา Multicollinearity ตรวจสอบจากค่า Variance Inflation Factor (VIF) พบว่าค่า VIF มากสุดอยู่ที่ 3.409 ซึ่งยังไม่เกิน 10 แสดงว่าไม่เกิดปัญหามาตรวัดวัดค่าซ้ำซ้อน

2.2 พิจารณาคัดเลือกมาตรวัดเข้าสมการ การสร้างโมเดลพยากรณ์ฯ มีวิธีพิจารณามาตรวัดเข้าสมการ ดังนี้

1) ทดสอบชุดของตัวแปรอิสระว่ามีความสัมพันธ์กับตัวแปรตามหรือไม่ (Over All Model Tests) พิจารณาจาก X^2 ที่สมการ (4)

$$X^2 = 2[\ln L_p - \ln L_0] \quad (4)$$

L_p คือ ค่า Log Likelihood ของโมเดลที่มีมาตรวัดจำนวนทั้งหมด 11 ตัวตามจำนวนมาตรวัด และ L_0 คือ ค่า Log Likelihood ของโมเดลที่ไม่มีมาตรวัดใดเลย สามารถพิจารณาจาก Sig. ของตาราง Model Fitting Information ของ SPSS หากน้อยกว่าระดับนัยสำคัญทางสถิติ 0.05 จะสรุปได้ว่ามาตรวัดมีความสัมพันธ์กับค่า Analyzability

เมื่อพิจารณาจากค่า Sig. เท่ากับ 0.00 ซึ่งน้อยกว่า 0.05 หมายความว่า มาตรวัดและค่า Analyzability มีความสัมพันธ์ต่อกัน

2) ทดสอบว่าตัวแปรอิสระแต่ละตัวมีผลต่อตัวแปรตามหรือไม่ (Examining the Coefficients) พิจารณาจาก Wald Statistic ซึ่งมีการแจกแจงแบบไคสแควร์ที่มี Degree Of Freedom เท่ากับ 1 ที่สมการ (5)

$$\text{Wald} = \left[\frac{B}{\text{SE}} \right]^2 \quad (5)$$

ค่า Wald statistic จะเป็นกำลังสองของอัตราส่วนระหว่างค่าสัมประสิทธิ์ของ Logit ที่ใช้บอกทิศทางและปริมาณผลกระทบต่อตัวแปรอิสระที่มีผลต่อการเกิดตัวแปรตาม (B) กับค่าความคลาดเคลื่อนมาตรฐานของสัมประสิทธิ์ (SE) ซึ่งสามารถพิจารณาจาก Sig. ของตาราง Parameter Estimates ของ SPSS หากน้อยกว่าระดับนัยสำคัญทางสถิติ 0.05 จะสรุปได้ว่ามาตรวัดมีผลต่อค่า Analyzability

เมื่อพิจารณาค่า Sig. พบว่ามีมาตรวัดที่มีค่า Sig. น้อยกว่า 0.05 อยู่ 2 ตัว ได้แก่ CBO มีค่าเป็น -0.239 ซึ่งมีค่าสถิติของ Wald คือ $[-0.239/0.085]^2 = 7.931$ ที่ Sig. = 0.005 และ LCOM มีค่าเป็น 7.755 ซึ่งมีค่าสถิติของ Wald คือ $[7.755/3.520]^2 = 4.854$ ที่ Sig. = 0.028 จะได้โมเดลตามสมการ (2) ของระดับชั้นของค่า Analyzability (Y) ในแต่ละระดับคุณภาพ 1-3 สมการ (6) – (8) ตามลำดับ

$$P(Y \leq 1) = 1/(1 + \exp(-((-6.189) - (-0.239)CBO - (7.755)LCOM))) \quad (6)$$

$$P(Y \leq 2) = P(Y \leq 2) = 1/(1 + \exp(-((-1.100) - (-0.239)CBO - (7.755)LCOM))) \quad (7)$$

$$P(Y \leq 3) = 1 \quad (8)$$

2.3 วิธีการทดสอบโมเดล เป็นวิธีการทดสอบความเหมาะสมของโมเดลหลังจากได้สมการเรียบร้อยแล้ว ดังนี้

1) ทดสอบสัมประสิทธิ์ของตัวแปรชนิดเดียวกันในโมเดลจะต้องเท่ากัน (Test Parallel Lines) เป็นการทดสอบโมเดลแต่ละระดับชั้นที่ได้จะต้องขนานกัน โดย β_j เป็นสัมประสิทธิ์ของตัวแปรอิสระตัวที่ i ในกลุ่มระดับชั้นที่ j โดยค่า $1 \leq i \leq m$ และ m คือ จำนวนตัวแปรของสมการถดถอย และค่า j เป็นระดับชั้นของค่าตัวแปรตาม ซึ่งสามารถพิจารณาได้จาก Sig. ของตาราง Test Of Parallel Lines ของ SPSS หากมากกว่าระดับนัยสำคัญทางสถิติ 0.05 จะสรุปได้ว่า สามารถใช้สัมประสิทธิ์ของมาตรวัดที่เหมือนกันในทุกสมการ

เมื่อพิจารณาค่า Sig. พบว่า เท่ากับ 0.311 ซึ่งมากกว่า 0.05 จึงสรุปได้ว่า สามารถใช้สัมประสิทธิ์ของมาตรวัดที่เหมือนกันในทุกสมการ

2) ทดสอบความเหมาะสมของโมเดลที่ได้ (Goodness Of Fit Measures) เป็นการวัดระดับของความสัมพันธ์ระหว่างเชิงเส้นที่สัมพันธ์กับตัวแปรหลายตัว พิจารณาได้จากสถิติ Pearson (X^2) สมการ (9)

$$X^2 = \sum \sum (O_{ij} - E_{ij})^2 / E_{ij} \quad (9)$$

O_{ij} คือ ความถี่ที่สังเกตได้ และ E_{ij} คือ ความถี่ที่คาดหวัง ซึ่งสามารถพิจารณาจาก Sig. ของตาราง Goodness-Of-Fit ของ SPSS หากระดับนัยสำคัญทางสถิติต่ำกว่า 0.05 จะสรุปได้ว่า โมเดลที่ใช้มีความเหมาะสม

เมื่อพิจารณาค่า Sig. พบว่า Sig. ของ Pearson เท่ากับ 0.00 ซึ่งน้อยกว่า 0.05 จึงสรุปได้ว่า โมเดลมีความเหมาะสม

3) ทดสอบความเหมาะสมของตัวแบบ (Strength Of Association) ตรวจสอบว่าโมเดลที่ได้สามารถอธิบายค่าตัวแปรตามได้ดีแค่ไหน พิจารณาจากค่าสัมประสิทธิ์การตัดสินใจ (R square หรือ R^2) เป็นค่าของความแปรปรวนของค่าตัวแปรตามที่สามารถอธิบายได้จากตัวแปรอิสระ ซึ่งถ้าคิดเป็นเปอร์เซ็นต์ให้คุณด้วย 100 จะใช้ R^2 ของ Nagelkerke (R^2_N) สมการ (10)

$$R^2_N = \frac{R^2_{CS}}{1 - L(B^{(0)})/n} \quad (10)$$

จากตาราง Pseudo R-square ของ SPSS พบว่าค่า R square แบบ Nagelkerke มีค่าเท่ากับ 0.694 ซึ่งมาตรวจวัดจะสามารถอธิบายความแปรผันของค่า Analyzability ได้ร้อยละ 69.4

สำหรับการนำโมเดลไปใช้ประเมินคุณภาพด้าน Analyzability ของคลาสมีขั้นตอน ดังนี้

- 1) วัดค่ามาตรวัด CBO และ LCOM ของคลาสที่ต้องการประเมินระดับคุณภาพ Analyzability
- 2) นำค่ามาตรวัดที่ได้ไปคำนวณหาความน่าจะเป็นสะสมของแต่ละระดับคุณภาพ ได้แก่ $P(Y \leq 1)$, $P(Y \leq 2)$ และ $P(Y \leq 3)$
- 3) คำนวณความน่าจะเป็นของคุณภาพแต่ละระดับ ดังนี้ ระดับคุณภาพ 1 หรือค่า $P(Y=1)$ จะเท่ากับ $P(Y \leq 1)$ ระดับคุณภาพ 2 หรือ ค่า $P(Y=2)$ จะมีเท่ากับ $P(Y \leq 2) - P(Y \leq 1)$ และระดับคุณภาพ 3 หรือ ค่า $P(Y=3)$ จะมีค่าเท่ากับ $1 - P(Y \leq 2)$
- 4) เลือกค่าความน่าจะเป็นสูงสุด ซึ่งจะระดับคุณภาพของคลาสนั้น

สรุปงานวิจัยและข้อเสนอแนะ

ในงานวิจัยนี้ได้สร้างโมเดลประเมินคุณภาพด้านความสามารถในการวิเคราะห์โค้ดจาวาระดับคลาสด้วยวิธีทางสถิติการถดถอยโลจิสติกแบบอันดับ ซึ่งพบว่ามาตรวัด Coupling Between Object (CBO) และ Lack Of Cohesion In Methods (LCOM) มีผลต่อคุณภาพด้านความสามารถในการวิเคราะห์เพื่อหาข้อผิดพลาดของซอร์สโค้ด นอกจากนั้นพบว่าโมเดลที่ได้มีความเหมาะสมและน่าเชื่อถือสามารถนำไปใช้พยากรณ์วิเคราะห์ค่าได้อย่างถูกต้อง เนื่องจากได้ผ่านการทดสอบเงื่อนไขการใช้การถดถอยโลจิสติกก่อนการสร้างโมเดล และได้มีกระบวนการในการพิจารณาตัวแปรอิสระเข้ามาในสมการอย่างเหมาะสมและมีการทดสอบความเหมาะสมของโมเดลระดับความเชื่อมั่น 95% แต่ตัวแปรอิสระสามารถอธิบายความแปรผันของตัวแปรตามได้ร้อยละ 69.4 ซึ่งในงานวิจัยในอนาคตสามารถเพิ่มจำนวนคลาสที่ใช้ในการสร้างโมเดลให้มากขึ้นเพื่อให้สามารถได้ค่าความแปรผันที่สูงขึ้น นอกจากนั้น

งานวิจัยที่น่าสนใจในอนาคตคือการทดลองใช้โมเดลประเมินคุณภาพด้าน Analyzability ที่นำเสนอไปใช้ประเมินคุณภาพด้าน Analyzability ของโค้ดที่พัฒนาด้วยภาษาโปรแกรมเชิงวัตถุอื่นที่ไม่ใช่ภาษาจาวา ว่าสามารถใช้ประเมินได้หรือไม่

เอกสารอ้างอิง

- กัลยา วานิชย์บัญชา. (2549). สถิติสำหรับงานวิจัย. (พิมพ์ครั้งที่ 2) กรุงเทพฯ: โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย.
- มะแอน ราโอบ. การวิเคราะห์การถดถอยโลจิสติก (logistic regression analysis). แหล่งข้อมูล: <http://www.saruthipong.com/port/document/299-705/299-705-7.pdf>. ค้นเมื่อวันที่ 1 มกราคม 2557.
- Antonellis, P., Antoniou, D., Kanellopoulos, Y., Makris, C., Theodoridis, E., Tjortjis, C., Tsirakis, N. (2007). A data mining methodology for evaluating maintainability according to ISO/IEC-9126 software engineering-product quality standard. Special Session on System Quality and Maintainability-SQM2007.
- Chidamber, S. R., Kemerer, C. F. (1994). A metrics suite for object oriented design. *Software Engineering, IEEE Transactions on*. 20(6): 476-493.
- Heitlager, I., Kuipers, T., Visser, J. (2007). A Practical Model for Measuring Maintainability. In: *Quality of Information and Communications Technology, 2007. QUATIC 2007. 6th International Conference*. 30-39.
- IBM. SPSS. Available: <http://www-01.ibm.com/software/analytics/spss/products/statistics/>. Retrieved 1 January 2014.
- JEdit Development Team. (2013). jEdit. Available: <http://www.jedit.org/>. Retrieved 9 March 2014.
- Kanellopoulos, Y., Antonellis, P., Antoniou, D., Makris, C., Theodoridis, E., Tjortjis, C., Tsirakis, N. (2010). Code quality evaluation methodology using the iso/iec 9126 standard. arXiv preprint arXiv:1007.5117.
- Khan, S. A., Khan, R. A. (2012). Analyzability Quantification Model of Object Oriented Design. *Procedia Technology*. 4(0): 536-542.
- Lincke, R., Lundberg, J., Lowe, W. (2008). Comparing software metrics tools. In: *The 2008 international symposium on Software testing and analysis*. USA. 131-142.
- Meananeatra, P., Kitcharoensup, T., Rattanaleadnusorn, E., Charoendouysil, B., Wisuttikul, T., Rongviriyapanish, S. (2013). Modeling code analyzability at method level in J2EE applications. In: *The 2013 20th Asia-Pacific Software Engineering Conference 2013*.
- Rizvi, S.W.A., Khan, R.A. (2010). Maintainability estimation model for object-oriented software in design phase (memood). arXiv preprint arXiv:1004.4447.
- Scientific Toolworks, Inc. Understand 3.1.716 tool. Available: <http://www.scitools.com/index.php>. Retrieved 1 January 2014.
- Sourceforge. Metrics 1.3.6 tool. Available: <http://metrics.sourceforge.net/>. Retrieved 1 January 2014.

