บทที่ 1
ที่มาและความสำคัญของปัญหา

1.1 บทนำ

ในโครงการพัฒนาซอฟต์แวร์ขนาดใหญ่ที่มีช่วงการพัฒนาและช่วงการบำรุงรักษาที่ยาวนาน นักพัฒนามักจะประสบปัญหาหลายประการในแง่ของการทำงานเป็นทีม เช่น ปัญหาการทำความเข้าใจพัฒนาการของซอฟต์แวร์ของนักพัฒนาที่เข้าร่วมทีมใหม่ ปัญหาการเปลี่ยนแปลงแก้ไขซอฟต์แวร์ในช่วงการบำรุงรักษาที่อาจเกิดกับทีมบำรุงรักษาที่ไม่ใช่ทีมเดียวกับทีมพัฒนา ปัญหาการเรียกใช้ซอฟต์แวร์ไลบรารี่ที่ผิดของนักพัฒนาที่เข้าร่วมทีมใหม่ที่อาจนำไปสู่การเกิดข้อผิดพลาด เป็นต้น ปัญหาต่างๆเหล่านี้สามารถตอบสนองได้โดยการประยุกต์ใช้เทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Association Rule on Software Archives) งานวิจัยในอดีตที่ศึกษาการประยุกต์เทคนิคดังกล่าวมักจะใช้ตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น (Support-Confidence Model) เป็นตัวแบบในการประเมินความน่าสนใจของกฎความสัมพันธ์ แต่ตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นมีข้อบกพร่องที่สำคัญ คือ การให้ผลลัพธ์ที่เป็นผลบวกลวงจำนวนมาก ต่อมาในปีค.ศ. 2008 Liu และคณะ (Liu et al., 2008) ได้เสนอตัวแบบในการประเมินความน่าสนใจของกฎความสัมพันธ์ใหม่ขึ้นมาและให้ชื่อว่าค่าสนับสนุน-ค่าความเชื่อมั่นใหม่ (Support-New confidence Model) เพื่อปรับปรุงข้อบกพร่องดังกล่าวของตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นโดยเฉพาะ ผู้วิจัยเห็นว่าการนำค่าสนับสนุน-ค่าความเชื่อมั่นใหม่มาประยุกต์ใช้กับการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์น่าจะสามารถเพิ่มประสิทธิภาพให้กับระบบให้คำแนะนำนักพัฒนาในระหว่างการพัฒนาซอฟต์แวร์ได้ ส่งผลให้นักพัฒนาสามารถทำงานเป็นทีมได้อย่างมีประสิทธิภาพมากขึ้นด้วย

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาเปรียบเทียบประสิทธิภาพการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ ด้วยตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น ตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นใหม่ของ Liu และคณะ (Liu et al., 2008) ในสถานการณ์ของการให้คำแนะนำนักพัฒนาใน 3 สถานการณ์ได้แก่ 1) สถานการณ์การนำทาง (Navigation) 2) สถานการณ์การป้องกันการเกิดข้อผิดพลาด (Error Prevention) และ 3) สถานการณ์การเปลี่ยนแปลงแก้ไขที่สมบูรณ์แล้ว (Closure)

ในนี้ผู้วิจัยจะนำเสนอความสำคัญของปัญหาในเบื้องต้นที่ประกอบไปด้วย ความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของการวิจัย ตัวแปรที่ศึกษา ประโยชน์ที่คาดว่าจะได้รับ ข้อจำกัดของงานวิจัยนี้ และนิยามของศัพท์สำคัญในงานวิจัยนี้

1.2 ความเป็นมาและความสำคัญของปัญหา

การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์เป็นเทคนิคหนึ่งที่ได้รับความนิยมและถูกนำไปประยุกต์ใช้กับข้อมูลหลายหลากแขนง หนึ่งในนั้นก็คือการประยุกต์ใช้กับข้อมูลซอฟต์แวร์อาร์ไคฟว์หรือข้อมูลการเปลี่ยนแปลงแก้ไขซอฟต์แวร์ในอดีตที่ได้มาจากระบบควบคุมการเปลี่ยนแปลงแก้ไข (Revision Control System, Version Control System) เพื่อประโยชน์ในการแก้ไขปัญหาต่างๆที่เกิดขึ้นกับนักพัฒนาในระหว่างการพัฒนาซอฟต์แวร์ ความสำคัญของระบบควบคุมการเปลี่ยนแปลงแก้ไขและปัญหาของนักพัฒนาในโครงการพัฒนาซอฟต์แวร์ขนาดใหญ่ถูกรวบรวมเอาไว้หัวข้อ 1.2.1 แต่การตอบสนองปัญหาเหล่านั้นด้วยการประยุกต์ใช้เทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ในอดีต เลือกใช้ตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นเป็นตัวแบบในการประเมินความน่าสนใจของกฎความสัมพันธ์ทั้งหมด ตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นเป็นตัวแบบที่ได้รับความนิยมมากแต่ในบางกรณีที่ประยุกต์ใช้กับข้อมูลบางประเภทก็สามารถทำให้เกิดผลลัพธ์ที่เป็นผลบวกลวงจำนวนมากได้ ปัญหาของค่าประเมินความน่าสนใจของกฎความสัมพันธ์ที่อาจมีผลกระทบมาถึงประสิทธิภาพของการนำไปประยุกต์ใช้รวบรวมเอาไว้ในหัวข้อ 1.2.2 ส่วนการประยุกต์ใช้เทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ของงานวิจัยในอดีตรวมถึงการประยุกต์ใช้กในระบบให้คำแนะนำนักพัฒนาในระหว่างการพัฒนาซอฟต์แวร์แสดงไว้ในหัวข้อ 1.2.3

1.2.1 ความสำคัญของระบบควบคุมการเปลี่ยนแปลงแก้ไขและปัญหาของนักพัฒนาในโครงการพัฒนาซอฟต์แวร์ขนาดใหญ่

เป็นที่ยอมรับกันว่าโลกธุรกิจทางด้านซอฟต์แวร์ในปัจจุบันมีการแข่งขันสูงมาก บริษัทที่จะอยู่ในตลาดได้จำเป็นอย่างยิ่งที่จะต้องมีเงินทุน บุคคลากรและกลยุทธ์ที่น่าสนใจ ฉะนั้นการสร้างความได้เปรียบทางการแข่งขันจำเป็นต้องมีกลยุทธ์ด้านกระบวนการที่มีประสิทธิภาพและมีมาตรฐานเป็นที่ยอมรับในระดับสากล เพื่อให้การผลิตซอฟต์แวร์มีคุณภาพ ตอบสนองความ

ต้องการและสร้างความพึงพอใจสูงสุดต่อลูกค้า มาตรฐานซีเอ็มเอ็มไอ (CMMI, Capability Maturity Model Integration) เป็นตัวแบบของการวัดระดับวุฒิภาวะ (Maturity) ความสามารถในการทำงานของบริษัท มาตรฐานซีเอ็มเอ็มไอที่ใช้ในปัจจุบันคือเวอร์ชั่น 2.1 ระดับวุฒิภาวะของมาตรฐานซีเอ็มเอ็มไอมีทั้งหมด 5 ระดับระดับวุฒิภาวะทั้งหมดประกอบด้วยกลุ่มกระบวน 22 กลุ่มกระบวนการ ในการบรรลุระดับวุฒิภาวะที่ 2 ของมาตรฐานซีเอ็มเอ็มไอ บริษัทจำเป็นต้องบรรลุเป้าหมายของกลุ่มกระบวนการทั้งหมด 7 กลุ่ม หนึ่งในนั้นคือกลุ่มกระบวนการการจัดการการตั้งค่าองค์ประกอบ (CM: Configuration Management) ซึ่งมีเป้าหมายเฉพาะเจาะจง (Specific Goal) ที่จำเป็นต้องบรรลุให้ได้ 3 เป้าหมาย และ 1 ใน 3 ของเป้าหมายเฉพาะเจาะจงนั้นคือ การติดตามและควบคุมการเปลี่ยนแปลงแก้ไข (Track and Control Changes) การบรรลุเป้าหมายข้อนี้นั้นจำเป็นต้องใช้เครื่องมือที่มีชื่อว่าระบบควบคุมการเปลี่ยนแปลงแก้ไข (Revision Control System, Version Control System) เข้ามาช่วย (Grune et al., 2006)

ระบบควบคุมการเปลี่ยนแปลงแก้ไข คือ ระบบที่ใช้ในจัดการการจัดเก็บ การค้นคืน การระบุและการผสานการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูลซอร์สโค้ดของโปรแกรมประยุกต์ และสารสนเทศสำคัญอื่นๆที่พัฒนาขึ้นมาโดยทีมอย่างเป็นอัตโนมัติ ในซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไขนั้นจะมีการบันทึกแฟ้มข้อมูลซอร์สโค้ดและแฟ้มข้อมูลบันทึก (Log Files) ที่บรรจุข้อมูลที่เกี่ยวข้องกับการเปลี่ยนแปลงแก้ไขอื่นๆ อาทิเช่น ซอร์สโค้ดส่วนใดที่ถูกแก้ไข นักพัฒนาแก้ไข วันเวลาบันทึกเวอร์ชั่นใหม่ของซอร์สโค้ด และหมายเหตุของการบันทึกเวอร์ชั่นใหม่ เป็นต้น แฟ้มข้อมูลซอร์สโค้ดและแฟ้มข้อมูลบันทึกทั้งหมดจะถูกเรียกรวมกันว่า ซอฟต์แวร์อาร์ไคฟ์ (Software Archives) (Zimmermann et al., 2004; Zimmermann et al., 2005)

ระบบควบคุมการเปลี่ยนแปลงแก้ไขที่ได้รับความนิยมและถูกนำไปใช้อย่างแพร่หลายกว่า 2 ทศวรรษ คือ ระบบควบคุมการเปลี่ยนแปลงแก้ไขที่มีชื่อว่า ระบบคอนเคอเรนท์เวอร์ชั่น (Concurrent Versions System) (O'Sullivan et al., 2009) ระบบคอนเคอเรนท์เวอร์ชั่นถูกสร้างขึ้นมาให้บูรณาการรวมกับไอดีอี (IDE: Integrated Development Environment) ทำให้นักพัฒนาสามารถบรรลุเป้าหมายการติดตามและควบคุมการเปลี่ยนแปลงแก้ไขได้ ในขณะที่กำลังพัฒนาซอฟต์แวร์ในขั้นตอนการพัฒนาซอฟต์แวร์ (Development Phase) ของวงจรชีวิตการพัฒนาซอฟต์แวร์ (Software Development Life Cycle) ได้

ปัญหาที่มักจะเกิดขึ้นกับนักพัฒนาในโครงการพัฒนาซอฟต์แวร์ขนาดใหญ่ที่ต้องมีการทำงานเป็นทีมในแง่ของการควบคุมและติดตามการเปลี่ยนแปลงแก้ไขมีหลายประการ เช่น ปัญหา

การเกิดขึ้นของการเชื่อมโยงกัน (Evolution coupling) ระหว่างคลาสหรือระหว่างไฟล์ที่ไม่สามารถดักจับได้ในช่วงของการออกแบบ (Design phase) (Gall et al., 1998; Bieman et al., 2003; Burch et al., 2005) ปัญหาการทำความเข้าใจพัฒนาการของซอฟต์แวร์ (Software Evolution) (Ball et al., 1997) ที่อาจเกิดกับนักพัฒนาที่เข้าร่วมทีมใหม่ ปัญหาการเปลี่ยนแปลงแก้ไขซอฟต์แวร์ในช่วงการบำรุงรักษา (Maintenance phase) ที่อาจเกิดกับทีมบำรุงรักษาที่ไม่ใช่ทีมเดียวกับทีมพัฒนา ปัญหาการเรียกใช้ซอฟต์แวร์ไลบรารี่ที่ผิดและนำไปสู่การเกิดข้อผิดพลาด (Li et al., 2005; Livshits et al., 2005; Williams et al., 2005) นอกจากปัญหาต่างๆข้างต้นแล้ว ในระหว่างการพัฒนาซอฟต์แวร์นั้นอาจทำให้เกิดความต้องการบางอย่างเกิดขึ้นด้วย เช่น ความต้องการนำรูปแบบการเรียกใช้ซอฟต์แวร์ไลบรารี่ (Software Libraries) ที่ถูกต้องกลับมาใช้ใหม่ (Michail, 2000) ความต้องการระบบให้คำแนะนำนักพัฒนาในระหว่างการพัฒนาซอฟต์แวร์ (Zimmermann et al., 2004; Zimmermann et al., 2005; Methanias et al., 2009) ปัญหาและความต้องการที่กล่าวมาข้างต้นนี้สามารถตอบสนองได้โดยการนำข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Software Archives) ที่ได้มาจากระบบคอนเคอเรนท์เวอร์ชั่นมาวิเคราะห์และสร้างวิธีการในการแก้ปัญหาและตอบสนองความต้องการดังกล่าวได้ ซึ่งจะกล่าวถึงในหัวข้อที่ 1.2.3 ต่อไป

1.2.2 ความสำคัญและปัญหาของกฎความสัมพันธ์และค่าประเมินความน่าสนใจของกฎความสัมพันธ์

ทุกครั้งที่ผู้ใช้เข้าไปใช้บริการเลือกซื้อหนังสือหรือสินค้าต่างๆ ภายในเว็บไซต์อเมซอนดอทคอม (Amazon.com) ผู้ใช้จะสามารถมองเห็นส่วนหนึ่งของหน้าเว็บไซต์ปรากฏข้อความที่ว่า "ลูกค้าหลายๆคนที่ซื้อหนังสือเล่มนี้ (หรือสินค้าชิ้นนี้) มักจะซื้อหนังสือ (หรือสินค้า) ...ด้วย" พร้อมกับแสดงรายการหนังสือ (หรือสินค้า) ที่มักจะถูกซื้อร่วมกันด้วย ข้อมูลสารสนเทศที่เว็บไซต์อเมซอนดอทคอมนำมาใช้เพื่อประโยชน์ในการเพิ่มยอดขายนี้เป็นข้อมูลสารสนเทศที่สร้างมาจากการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์ (Association Rules Discovery) ทั้งสิ้น การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลการซื้อของลูกค้านั้นยังสามารถนำไปประยุกต์ใช้ในการออกแบบแค็ตตาล็อกสินค้า การขายสินค้าแขวน การออกแบบรายการส่งเสริมการขาย การจัดวางสินค้าภายในร้าน การแบ่งกลุ่มลูกค้าตามรูปแบบของพฤติกรรมการซื้อสินค้า เป็นต้น (Agrawal et al., 1994) นอกจากนั้นแล้วในตลอดช่วงทศวรรษที่ผ่านมาการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์นี้ยังมีบทบาทที่สำคัญในการค้นหารูปแบบความสัมพันธ์ที่มีคุณค่าในข้อมูลประเภทอื่นๆอีก เช่น ข้อมูลเครือข่ายโทรคมนาคม

ข้อมูลการจัดการความเสี่ยง ข้อมูลการควบคุมคลังสินค้า และข้อมูลทางพันธุกรรมของสิ่งมีชีวิต เป็นต้น (Kotsiantis et al., 2006)

แนวคิดของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์นี้ถูกนำเสนอขึ้นมาครั้งแรกในปีค.ศ. 1993 โดย Agrawal และคณะ (Agrawal et al., 1993) ต่อมาในปีค.ศ. 1994 Agrawal และคณะ (Agrawal et al., 1994) ได้นำเสนอขั้นตอนวิธีในการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์ใหม่ขึ้นมาชื่อว่าขั้นตอนวิธีอพริโอริ (Apriori Algorithm) นอกจากนั้น Agrawal และคณะ (Agrawal et al., 1993) ยังได้นำเสนอตัวแบบของการประเมินระดับความตรงประเด็นหรือระดับความน่าสนใจของกฎความสัมพันธ์ตัวแบบแรกและดั้งเดิมคือตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น (Support-Confidence Model) ซึ่งใช้ค่า 2 ค่าในการประเมิน คือ ค่าสนับสนุน (Support) และค่าความเชื่อมั่น (Confidence) ของกฎความสัมพันธ์ ในตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นนี้ ค่าสนับสนุนถูกใช้ในการคัดกรองรายการที่มีความถี่สูงออกมา และค่าความเชื่อมั่นจะถูกใช้เป็นค่าที่วัดระดับความน่าสนใจของกฎความสัมพันธ์ หลังจากนั้นต่อมามีงานวิจัยหลายงานวิจัยออกมาเสนอค่าประเมินค่าอื่นๆ ที่ใช้ในการประเมินความน่าสนใจของกฎความสัมพันธ์แทนการใช้ค่าความเชื่อมั่น ผู้วิจัยรวบรวมงานวิจัยที่เสนอค่าประเมินความน่าสนใจของกฎความสัมพันธ์ที่ได้รับความนิยมและถูกนำไปประยุกต์กับต่างๆ อย่างละเอียดไว้ในบทที่ 2 ซึ่งสามารถสรุปได้ดังตารางต่อไปนี้

กำหนดให้ $P(X)$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

$P(\overline{X})$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่ไม่มีรายการ X ในฐานข้อมูล

$P(X \text{ and } Y)$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และ Y ในฐานข้อมูล

$P(\overline{X} \text{ and } \overline{Y})$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่ไม่มีทั้งรายการ X และ Y ในฐานข้อมูล

$P(X \text{ and } \overline{Y})$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และไม่รายการ Y ในฐานข้อมูล

ตารางที่ 1-1 แสดงตารางสรุปค่าประเมินความน่าสนใจของกฎความสัมพันธ์

ชื่อค่าประเมินฯ	สมการคำนวณ	อ้างอิง
ค่าสนับสนุน (Support)	$\text{Support}(X \rightarrow Y) = P(X \text{ and } Y)$	(Agrawal et al., 1993)
ค่าความเชื่อมั่น (Confidence)	$\text{Conf}(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(X)}$	(Agrawal et al., 1993)
ค่าคอนวิคชั่น (Conviction)	$\text{Conviction}(X \rightarrow Y) = \frac{P(X)P(\overline{Y})}{P(X \text{ and } \overline{Y})}$	(Brin et al., 1997)
ค่าลิฟท์ (Lift)	$\text{Lift}(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(X)P(Y)}$	(Brin et al., 1997)
ค่าเลฟเวอเรจ (Leverage)	$\text{Leverage}(X \rightarrow Y) = P(X \text{ and } Y) - P(X)P(Y)$	(Piatetsky-Shapiro et al., 1991)
ค่าคัฟเวอเรจ (Coverage)	$\text{Coverage}(X \rightarrow Y) = P(X)$	(Michael., 2009)
ค่าสหสัมพันธ์ (Correlation)	$\text{Corr}(X \rightarrow Y) = \frac{(P(X \text{ and } Y) - P(X)P(Y))}{\sqrt{P(X)P(Y)P(1-P(X))(1-P(Y))}}$	(Sheikh et al., 2004)
ค่าอัตราส่วนออดส์ (Odds Ratio)	$\text{Odds}(X \rightarrow Y) = \frac{(P(X \text{ and } Y)\ P(\overline{X} \text{ and } \overline{Y}))}{(P(X \text{ and } \overline{Y})\ P(\overline{X} \text{ and } Y))}$	(Sheikh et al., 2004)

ต่อมาปีค.ศ. 2008 Liu และคณะ (Liu et al., 2008) ได้นำเสนอข้อบกพร่องประการหนึ่งของการใช้ตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นและการใช้ค่าความเชื่อมั่นเป็นค่าประเมินความน่าสนใจของกฎความสัมพันธ์ โดยการยกตัวอย่างฐานข้อมูลการซื้อสินค้าของลูกค้าในกรณีที่ทำให้การใช้ค่าความเชื่อมั่นเป็นค่าประเมินความน่าสนใจของกฎความสัมพันธ์มีผลลัพธ์ที่ออกมาเป็นกฎความสัมพันธ์ที่มีเซตรายการที่มาก่อนมีความสัมพันธ์เชิงลบกับเซตรายการที่ตามมา กล่าวคือทรานแซคชั่นส่วนใหญ่ถ้ามีเซตรายการที่มาก่อนมักจะไม่มีเซตรายการที่ตามมาของกฎนั้นนั่นเอง หรือก็คือได้กฎความสัมพันธ์ที่เป็นผลบวกลวง (False Positive) นั่นเอง ด้วยสาเหตุนี้ Liu และคณะ (Liu et al., 2008) จึงได้นำเสนอค่าประเมินความน่าสนใจของกฎความสัมพันธ์ใหม่ขึ้นมาและให้ชื่อว่าค่าความเชื่อมั่นใหม่ (New Confidence) พร้อมกับพิสูจน์ว่าค่าความเชื่อมั่นใหม่นี้ไม่ขัดแย้งกับค่าสหสัมพันธ์และค่าความเชื่อมั่นเดิมซึ่งเป็นค่าสถิติ นอกจากนั้นยังได้แสดงตัวอย่างของฐานข้อมูลทรานแซคชั่นสมมุติชุดหนึ่งขึ้นมาเพื่อพิสูจน์ว่าค่าความเชื่อมั่นใหม่สามารถลดการเกิดกฎความสัมพันธ์ที่เป็นผลบวกลวงด้วย ค่าความเชื่อมั่นใหม่สามารถคำนวณได้จากสูตรดังต่อไปนี้

กำหนดให้ $P(X)$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

$P(\overline{X})$ คือ ค่าความน่าจะเป็นในการไม่พบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

$P(X \text{ and } Y)$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และรายการ Y

ในฐานข้อมูล

$P(X \text{ and } \bar{Y})$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และไม่รายการ Y ในฐานข้อมูล

$$NConf(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(Y)} - \frac{P(X \text{ and } \bar{Y})}{P(\bar{Y})}$$

ในงานวิจัยของ Liu และคณะ (Liu et al., 2008) ที่ได้เสนอค่าความเชื่อมั่นใหม่ข้างต้นนั้น Liu และคณะได้ทำการเปรียบเทียบความสามารถของค่าความเชื่อมั่นใหม่กับค่าประเมินความน่าสนใจของกฎความสัมพันธ์อื่นๆ ทั้งหมด 8 ค่าคือ ค่าสนับสนุน (Support), ค่าความเชื่อมั่น (Confidence), ค่าคอนวิคชั่น (Conviction), ค่าลิฟท์ (Lift), ค่าเลฟเวอเรจ (Leverage), ค่าคัฟเวอเรจ (Coverage), ค่าสหสัมพันธ์ (Correlation) และ ค่าอัตราส่วนออดส์ (Odds Ratio) โดยใช้ฐานข้อมูลทรานแซคชั่นสมมุติขนาด 10 ทรานแซคชั่น ผลของการเปรียบเทียบคือ ค่าความเชื่อมั่นใหม่สามารถบ่งบอกทิศของความสัมพันธ์ได้อย่างถูกต้องและสอดคล้องกับค่าเลฟเวอเรจและค่าสหสัมพันธ์ แต่ค่าความเชื่อมั่นใหม่นั้นสามารถระบุความแตกต่างของความน่าสนใจของกฎความสัมพันธ์ 2 กฎความสัมพันธ์ใดๆที่ค่าเลฟเวอเรจและค่าสหสัมพันธ์ไม่สามารถระบุได้ (กล่าวคือกฎความสัมพันธ์ 2 กฎที่คำนวณค่าค่าเลฟเวอเรจหรือค่าสหสัมพันธ์ได้เท่ากันทั้ง 2 กฎ แต่ค่าความเชื่อมั่นใหม่ให้ค่าที่แตกต่างกันระหว่าง 2 กฎ) ส่วนค่าประเมินความน่าสนใจของกฎความสัมพันธ์อื่นๆให้ค่าที่ขัดแย้งกับค่าสหพันธ์ (กล่าวคือเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์นั้นมีความสัมพันธ์เชิงลบต่อกันแต่กลับให้ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ที่สูงออกมา)

จากงานวิจัยในอดีตที่ทำการเสนอค่าประเมินความน่าสนใจของกฎความสัมพันธ์ต่างๆที่กล่าวไปข้างต้น แต่ละค่านั้นก็แสดงคุณสมบัติเฉพาะตัวที่แตกต่างกัน ผลลัพธ์ของกฎความสัมพันธ์ที่ได้ออกมาก็แตกต่างกันออกไป ผู้วิจัยจึงสนใจที่จะทำการเปรียบเทียบความสามารถของแต่ละค่าประเมินความน่าสนใจของกฎความสัมพันธ์ ผู้วิจัยจึงทบทวนและรวบรวมงานวิจัยที่เสนอคุณสมบัติต่างๆที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ควรจะมี และทำการเปรียบเทียบความสามารถของค่าประเมินความน่าสนใจของกฎความสัมพันธ์เหล่านั้นด้วยคุณสมบัติที่ควรจะมีทั้งหมด คุณสมบัติของค่าประเมินความน่าสนใจของกฎความสัมพันธ์ทั้งหมด 16 คุณสมบัติอธิบายอย่างละเอียดไว้ในบทที่ 2 หัวข้อ 2.5 และสามารถสรุปได้ดังนี้

- คุณสมบัติ 3 ข้อของ Piatetsky-Shapiro และคณะ (Piatetsky-Shapiro, 1991)
- คุณสมบัติ 1 ข้อของ Major และ Mangano (Major and Mangano, 1995) เพิ่มเติมจากของ Piatetsky-Shapiro และคณะ
- คุณสมบัติ 5 ข้อของ Tan และคณะ (Tan et al, 2002)
- คุณสมบัติ 5 ข้อของ Lenca และคณะ (Lenca et al, 2004)
- คุณสมบัติ 2 ข้อของ Geng และ Hamilton (Geng and Hamilton, 2006)

คุณสมบัติที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ควรจะมีทั้งหมด 16 ข้อข้างต้น คุณสมบัติที่ได้รับการยอมรับและถูกอ้างอิงถึงโดยงานวิจัยต่างๆ (Freitas, 1999; Major and Mangano, 1995; Mcgarry, 2005; Geng and Hamilton, 2006; Liu et al., 2008; Heravi, 2009) มากที่สุดคือคุณสมบัติ P1 P2 และ P3 ของ Piatetsky-Shapiro และคณะ (Piatetsky-Shapiro, 1991) และคุณสมบัติ P4 ของ Major และ Mangano (Major and Mangano, 1995)

เนื่องจากงานวิจัยของ Liu และคณะในปี 2008 (Liu et al., 2008) ได้ทำการพิสูจน์คุณสมบัติของค่าความเชื่อมั่นใหม่ไว้ทั้งหมดเพียง 5 คุณสมบัติคือ คุณสมบัติ P1 P2 P3 O1 และ O2 เท่านั้น ดังนั้นผู้วิจัยจึงพิสูจน์คุณสมบัติ P4 O3 O4 O5 Q1 Q2 Q3 S1 และ S2 ของค่าความเชื่อมั่นใหม่อย่างละเอียดและแสดงไว้ในบทที่ 2 หัวข้อ 2.5 ผลของการพิสูจน์คุณสมบัติของค่าความเชื่อมั่นใหม่แสดงไว้ในตารางที่ 2-4 ข้างต้น

การเปรียบเทียบในตารางที่ 2-4 แสดงให้เห็นว่าค่าความเชื่อมั่นใหม่มีคุณสมบัติที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ควรจะมี 10 คุณสมบัติจากทั้งหมด 14 คุณสมบัติซึ่งมากกว่าค่าประเมินความน่าสนใจของกฎความสัมพันธ์อื่นๆ โดยเฉพาะอย่างยิ่งการมีคุณสมบัติ O3 ของค่าความเชื่อมั่นใหม่จะทำให้การนำค่าความเชื่อมั่นใหม่ไปใช้นั้นจะสามารถขจัดการเกิดกฎความสัมพันธ์ที่มีเซตรายการที่มาก่อนและเซตรายการที่ตามที่มีความสัมพันธ์เชิงลบออกไปได้ ผู้วิจัยจึงเชื่อว่าถ้านำค่าความเชื่อมั่นใหม่นี้ไปประยุกต์ใช้กับการค้นหากฎความสัมพันธ์กับข้อมูลประเภทต่างๆรวมถึงข้อมูลซอฟต์แวร์อาร์ไคฟ์ แล้วน่าจะทำให้กฎความสัมพันธ์ที่ได้มาเป็นกฎความสัมพันธ์ที่น่าสนใจและช่วยลดการเกิดกฎความสัมพันธ์ที่เป็นผลบวกลวง (False Positive) ได้

1.2.3 ความสำคัญและปัญหาของการประยุกต์ใช้เทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์

ในช่วงต้นของการคิดค้นและพัฒนาแนวคิดการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์นั้น การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์ถูกพัฒนามาเพื่อการค้นหารูปแบบความสัมพันธ์ของพฤติกรรมการซื้อสินค้าของลูกค้าจากฐานข้อมูลรายการซื้อสินค้าขนาดใหญ่ ต่อจากนั้นมาไม่นานเริ่มมีนักวิจัยหลายคณะนำการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์มาประยุกต์ใช้กับข้อมูลประเภทต่างๆมากมาย หนึ่งในนั้นคือข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Software Archive) ที่ได้จากระบบคอนเคอเรนท์เวอร์ชั่นในขั้นตอนการพัฒนาซอฟต์แวร์ (Development Phase) ในวงจรชีวิตการพัฒนาซอฟต์แวร์ (Software Development Life Cycle) คณะนักวิจัยเหล่านั้นนำข้อมูลซอฟต์แวร์อาร์ไคฟว์มาวิเคราะห์ในรูปแบบต่างๆกันแบ่งตามการนำไปใช้ ดังต่อไปนี้

1) การวิเคราะห์เพื่อความเข้าใจพัฒนาการของการพัฒนาซอฟต์แวร์ (Ball et al., 1997)
2) การวิเคราะห์เพื่อดักจับพัฒนาการของการเชื่อมโยงกัน (Gall et al.,1998; Bieman et al., 2003; Zimmermann et al., 2004)
3) การวิเคราะห์เพื่อเปิดเผยรูปแบบการเรียกใช้งานซอฟต์แวร์ไลบรารี่ (Michail, 1999; Michail, 2000; Li et al., 2005; Livshits et al., 2005; Williams et al., 2005)
4) การวิเคราะห์เพื่อสร้างคำแนะนำในการเปลี่ยนแปลงแก้ไข (Zimmermann et al., 2005; Methanias et al., 2009)

รายละเอียดของแต่ละงานวิจัยแสดงในหัวข้อ 2.7 จากงานวิจัยที่นำข้อมูลซอฟต์แวร์อาร์ไคฟว์มาวิเคราะห์ทั้งหมด มีคณะวิจัยของ Li และคณะวิจัยของ Livshits (Li et al., 2005; Livshits et al., 2005) ได้แสดงให้เห็นว่าการประยุกต์เทคนิคการทำเหมืองข้อมูลด้วยกฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์เป็นเทคนิคที่ค่อนข้างมีประสิทธิภาพแต่ในบางกรณีก็สามารถทำให้เกิดผลลัพธ์ของการค้นหาที่เป็นผลบวกลวง (False Positive) เป็นจำนวนมากได้

ในปีค.ศ. 2005 Zimmermann และคณะ (Zimmermann et al., 2005) ได้เสนอการประยุกต์ใช้การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์อีกรูปแบบหนึ่ง คือการสร้างคำแนะนำในการเปลี่ยนแปลงแก้ไขให้กับนักพัฒนาในระหว่างขั้นตอนการพัฒนาซอฟต์แวร์ ในงานวิจัยนี้ Zimmermann และคณะเสนอว่าระบบสร้าง

คำแนะนำนักพัฒนานั้นควรจะสามารถให้คำแนะนำแก่นักพัฒนาทั้งหมด 3 สถานการณ์คือ 1) สถานการณ์การนำทาง (Navigation) คือ สถานการณ์ที่นักพัฒนามีการเปลี่ยนแปลงแก้ไขที่เอนทิตี้หนึ่งแล้ว ระบบจะให้คำแนะนำกับนักพัฒนาให้แก้ไขเอนทิตี้ใดต่อไป 2) สถานการณ์การป้องกันการเกิดข้อผิดพลาด (Error Prevention) คือ สถานการณ์ที่นักพัฒนามีการเปลี่ยนแปลงแก้ไขที่เอนทิตี้หลายๆเอนทิตี้ต่อเนื่องกันแต่ยังขาดการเปลี่ยนแปลงแก้ไขเอนทิตี้อีกหนึ่งเอนทิตี้จึงจะสมบูรณ์ ระบบจะให้คำแนะนำกับนักพัฒนาให้แก้ไขเอนทิตี้ที่เหลือนั่น และ 3) สถานการณ์การเปลี่ยนแปลงแก้ไขที่สมบูรณ์แล้ว (Closure) คือ สถานการณ์ที่นักพัฒนามีการเปลี่ยนแปลงแก้ไขที่เอนทิตี้หลายๆเอนทิตี้ต่อเนื่องกันจนสมบูรณ์แล้ว ระบบจะไม่ควรให้คำแนะนำที่เป็นผลบวกลวง (False Positive) ออกมาแก่นักพัฒนา งานวิจัยของ Zimmermann และคณะ (Zimmermann et al., 2005) ได้ทำการทดสอบประสิทธิภาพของการให้คำแนะนำในรูปแบบต่างๆหลายรูปแบบกับข้อมูลซอฟต์แวร์อาร์ไคฟ์วของโครงการพัฒนาซอฟต์แวร์แบบโอเพนซอร์ส (Open Source) และข้อสรุปของการทดสอบได้แนะนำสิ่งที่เป็นประโยชน์ต่อการต่อยอดการประยุกต์ใช้การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์วได้เป็นอย่างดี เช่น การให้คำแนะนำในระดับเอนทิตี้ละเอียด (ตัวแปร เมธอดหรือฟังก์ชั่น) ให้ประสิทธิภาพไม่ต่างกับการให้คำแนะนำในแฟ้มข้อมูลหรือคลาส การให้คำแนะนำนักพัฒนามีประสิทธิภาพมากสำหรับการเปลี่ยนแปลงแก้ไขในช่วงการบำรุงรักษา (Maintenance Phase) (เน้นที่การแก้ไข (alter) มากกว่าการเพิ่ม (add to) กับการลบ (delete from)) เป็นต้น นอกจากนั้นผู้วิจัยสังเกตเห็นว่าผลการทดสอบประสิทธิภาพในงานวิจัยของ Zimmermann และคณะ (Zimmermann et al., 2005) ยังให้ประสิทธิภาพไม่สูงเท่าที่ควร

จากความสำคัญและปัญหาที่กล่าวไปทั้งหมดในหัวข้อ 1.2.1 ถึง 1.2.3 ข้างต้นผู้วิจัยเห็นว่าการเพิ่มประสิทธิภาพให้กับระบบให้คำแนะนำนักพัฒนาในระหว่างการพัฒนาซอฟต์แวร์ของไอดีอีนั้นควรเพิ่มประสิทธิภาพโดยการปรับปรุงขั้นตอนวิธีในการค้นหากฎความสัมพันธ์ให้ดีขึ้นและลดจำนวนของการเกิดผลบวกลวงลง ผู้วิจัยจึงคาดว่าการทำเหมืองข้อมูลด้วยเทคนิคการ*ค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์วด้วยตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นใหม่ของ Liu และคณะ (Liu et al., 2008) นั้นน่าจะมีประสิทธิภาพที่ดีกว่าการทำเหมืองข้อมูลด้วย*เทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์วด้วยตัวแบบค่าสนับสนุน-ค่าความ

เชื่อมั่นเดิมจากการแสดงคุณสมบัติที่ค่าประเมินความน่าสนใจควรจะมีมากที่สุดและโดยเฉพาะอย่างยิ่งการแสดงคุณสมบัติ O3

โดยทั่วไปแล้วระบบให้คำแนะนำนักพัฒนาในระหว่างการพัฒนาซอฟต์แวร์ของไอดีอีจะทำหน้าที่หลัก 2 ประการคือ 1) การชี้นำนักพัฒนาว่าควรเปลี่ยนแปลงแก้ไขส่วนใดต่อเมื่อมีการเปลี่ยนแปลงแก้ไขส่วนนี้แล้ว และ 2) การแจ้งเตือนนักพัฒนาก่อนการบันทึกว่ายังทำการแก้ไขเปลี่ยนแปลงไม่สมบูรณ์เพื่อป้องกันการเกิดข้อผิดพลาด (error) นอกจากหน้าที่ 2 ประการนี้แล้วสิ่งที่สำคัญอีกประการหนึ่งคือระบบให้คำแนะนำต้องไม่มีการให้คำแนะนำใดๆออกมาถ้าการเปลี่ยนแปลงแก้ไขทั้งหมดสมบูรณ์ดีแล้วด้วย (Zimmermann et al., 2005) ดังนั้นในการทดสอบประสิทธิภาพของระบบให้คำแนะนำนักพัฒนาในระหว่างการพัฒนาซอฟต์แวร์ของไอดีอีควรทดสอบสถานการณ์ของการให้คำแนะนำนักพัฒนาต่างๆกัน 3 สถานการณ์ได้แก่ 1) สถานการณ์การนำทาง (Navigation) 2) สถานการณ์การป้องกันการเกิดข้อผิดพลาด (Error Prevention) 3) สถานการณ์การเปลี่ยนแปลงแก้ไขที่สมบูรณ์แล้ว (Closure)

ดังนั้นผู้วิจัยจึงต้องการศึกษาเปรียบเทียบประสิทธิภาพการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์ด้วยตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น (Support-Confidence Model) ดั้งเดิมกับการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์ด้วยตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นใหม่ของ Liu และคณะ (Liu et al., 2008) ในสถานการณ์ของการให้คำแนะนำนักพัฒนาต่างๆกัน 3 สถานการณ์

1.3 วัตถุประสงค์ของงานวิจัย

1. เปรียบเทียบประสิทธิภาพการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์ด้วยตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น (Support-Confidence Model) ดั้งเดิมกับการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์ด้วยตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นใหม่ของ Liu และคณะ (Liu et al., 2008) โดยที่ประสิทธิภาพของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับ

ข้อมูลซอฟต์แวร์อาร์ไคฟว์นั้นสามารถแบ่งออกได้เป็นประสิทธิภาพใน 3 สถานการณ์ของการพัฒนาซอฟต์แวร์ดังนี้

- สถานการณ์การนำทาง (Navigation)
- สถานการณ์การป้องกันการเกิดข้อผิดพลาด (Error Prevention)
- สถานการณ์การเปลี่ยนแปลงแก้ไขที่สมบูรณ์แล้ว (Closure)

1.4 ขั้นตอนโดยสรุปของการทำวิจัย

1. ศึกษารายละเอียดเกี่ยวกับการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ที่มีอยู่ในปัจจุบัน
2. ศึกษารายละเอียดเกี่ยวกับการประเมินระดับความน่าสนใจของกฎความสัมพันธ์ (Interestingness Measure of Association Rules) ในการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์ด้วยตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นใหม่ของ Liu และคณะ (Liu et al., 2008) และตัวแบบต่างๆที่มีอยู่ในปัจจุบัน
3. ออกแบบเครื่องมือทดสอบต่างๆตามที่ได้ศึกษา
4. พัฒนาเครื่องมือทดสอบตามที่ได้ออกแบบไว้
5. ทดสอบการทำงานของเครื่องมือที่พัฒนา
6. ประเมินการทำงานของเครื่องมือ
7. วิเคราะห์ผลการทดลองและสำรวจข้อมูลเพิ่มจากผลการทดลอง
8. จัดทำเอกสารสรุปงานวิจัย และข้อเสนอแนะ

1.5 ตัวแปรที่ศึกษา

1. **ตัวแปรอิสระ** (Independent variables)

งานวิจัยนี้สนใจว่าการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ด้วยตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นใหม่ของ Liu และคณะ (Liu et al., 2008) สามารถเพิ่มประสิทธิภาพของระบบให้คำแนะนำนักพัฒนาในระหว่างการพัฒนาซอฟต์แวร์

ของไอดีอี (IDE: Integrated Development Environment) ได้หรือไม่ ดังนั้นตัวแปรต้นของการศึกษานี้ก็คือตัวแบบของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์ ซึ่งงานวิจัยนี้จะศึกษาเปรียบเทียบประสิทธิภาพการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์ทั้งหมด 2 ตัวแบบ ดังนี้

1) การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์ด้วยตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น (Support-Confidence Model)
2) การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์ด้วยตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นใหม่ของ Liu และคณะ (Support-New Confidence Model) (Liu et al., 2008)

โดยในงานวิจัยนี้จะเปรียบเทียบประสิทธิภาพของทั้ง 2 ตัวแบบข้างต้นในสถานการณ์ที่ต่างกัน 3 สถานการณ์ คือ สถานการณ์การนำทาง (Navigation) สถานการณ์การป้องกันการเกิดข้อผิดพลาด (Error Prevention) และสถานการณ์การเปลี่ยนแปลงแก้ไขที่สมบูรณ์แล้ว (Closure) เช่นเดียวกับงานวิจัยของ Zimmermann และคณะในปี ค.ศ. 2005 (Zimmermann et al., 2005) และงานวิจัยของ Methanias และคณะในปี ค.ศ. 2009 (Methanias et al., 2009)

2. **ตัวแปรตาม** (Dependent variables)

ตัวแปรตามของงานวิจัยนี้ คือ ประสิทธิภาพการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์ด้วยตัวแบบต่างๆ การเปรียบเทียบประสิทธิภาพของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์จะพิจารณาจากคำแนะนำสำหรับนักพัฒนาที่ถูกสร้างมาจากกฎความสัมพันธ์ที่ได้มานั้นมีความถูกต้องแม่นยำในการทำนายและให้คำแนะนำในระหว่างการพัฒนาซอฟต์แวร์มากน้อยเพียงใด โดยการวัดประสิทธิภาพของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์ในแต่ละสถานการณ์มีวิธีการในการประเมินที่แตกต่างกันออกไปดังนี้

- ในสถานการณ์ *การนำทาง (Navigation)* สามารถประเมินประสิทธิภาพจากค่าเอฟเมสเซอร์ (F-measure) ที่คำนวณมาจากค่าความถูกต้อง (Precision) และค่าเรียกคืน (Recall) (Methanias et al., 2009) รายละเอียดและวิธีการคำนวณค่าเอฟเมสเซอร์ ค่าความถูกต้องและค่าเรียกคืนสำหรับการทำเหมืองข้อมูลด้วย

เทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์วนั้นได้กล่าวเอาไว้ในบทที่ 2

- ในสถานการณ์ *การป้องกันการเกิดข้อผิดพลาด (Error Prevention)* สามารถประเมินประสิทธิภาพจากค่าเอฟเมสเซอร์ (F-measure) ที่คำนวณมาจากค่าความถูกต้อง (Precision) และค่าเรียกคืน (Recall) (Methanias et al., 2009) รายละเอียดและวิธีการคำนวณค่าเอฟเมสเซอร์ ค่าความถูกต้องและค่าเรียกคืนสำหรับการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์วนั้นได้กล่าวเอาไว้ในบทที่ 2
- ในสถานการณ์ *การเปลี่ยนแปลงแก้ไขที่สมบูรณ์แล้ว (Closure)* สามารถประเมินประสิทธิภาพจากค่าผลสะท้อนกลับ (Feedback) (Zimmermann et al., 2005; Methanias et al., 2009) รายละเอียดและวิธีการคำนวณค่าผลสะท้อนกลับสำหรับการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์วนั้นได้กล่าวเอาไว้ในบทที่ 2

3. **ตัวแปรควบคุม**

ตัวแปรควบคุมกับการเปรียบเทียบประสิทธิภาพการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์ มีทั้งหมด 2 ตัวแปร ได้แก่

1) ข้อสอบถาม สำหรับงานวิจัยนี้ คือ เซตที่ประกอบไปด้วยเซตเหตุการณ์การเปลี่ยนแปลงแก้ไขและเซตผลลัพธ์ที่คาดไว้ โดยจะมีข้อสอบถามทั้งหมด 3 แบบสำหรับ 3 สถานการณ์ที่แตกต่างกันคือสถานการณ์การนำทาง สถานการณ์การป้องกันข้อผิดพลาด และสถานการณ์การเปลี่ยนแปลงแก้ไขที่สมบูรณ์แล้ว (Zimmermann et al., 2005; Methanias et al., 2009)
2) เครื่องมือที่ใช้ในงานวิจัย ประกอบด้วยเครื่องมือทั้งหมด 5 เครื่องมือได้แก่ เครื่องมือจัดเตรียมข้อมูลเพื่อการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟ์ เครื่องมือสร้างข้อสอบถามสำหรับการทดสอบ 3 สถานการณ์ เครื่องมือการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์ทั้ง 2 ตัวแบบสำหรับ 3 สถานการณ์ เครื่องมือสร้างเซตของคำแนะนำสำหรับเหตุการณ์ และเครื่องมือประเมินผลการทดสอบ

1.6 ขอบเขตของการวิจัย

1. ข้อมูลซอฟต์แวร์อาร์ไคฟว์ที่นำมาใช้ในการศึกษาเป็นข้อมูลซอฟต์แวร์อาร์ไคฟว์ของโครงการพัฒนาซอฟต์แวร์ที่พัฒนาด้วยภาษาซีพลัสพลัส (C++) เท่านั้น
2. ข้อมูลซอฟต์แวร์อาร์ไคฟว์ที่นำมาใช้ในการศึกษาเป็นข้อมูลซอฟต์แวร์อาร์ไคฟว์ที่มาจากซอฟต์แวร์ควบคุมการแก้ไขปรับปรุง (Revision Control) ที่ชื่อว่าระบบคอนเคอเรนท์เวอร์ชั่น (Concurrent Version System) เท่านั้น
3. คำแนะนำสำหรับนักพัฒนาในระหว่างการพัฒนาซอฟต์แวร์ในงานวิจัยนี้ หมายถึง คำแนะนำที่ได้มาจากการค้นหารูปแบบของการเปลี่ยนแปลงแก้ไขในการพัฒนาซอฟต์แวร์ที่เกิดขึ้นบ่อยในอดีตเท่านั้น ไม่รวมถึงรูปแบบของการเปลี่ยนแปลงแก้ไขที่เกิดขึ้นไม่บ่อยแต่มีความสำคัญมาก
4. คำแนะนำสำหรับนักพัฒนาในระหว่างการพัฒนาซอฟต์แวร์ในงานวิจัยนี้ สนใจเพียงการเปลี่ยนแปลงแก้ไขที่ควรจะอยู่ในทรานแซคชั่นเดียวกันเท่านั้น ไม่สนใจลำดับของการเปลี่ยนแปลงแก้ไข
5. ทรานแซคชั่นของการเปลี่ยนแปลงแก้ไขที่นำมาทดสอบในงานวิจัยนี้ ไม่รวมทรานแซคชั่นที่ถูกระบุว่าเป็นสิ่งแปลกปลอม 2 ประเภทคือ ทรานแซคชั่นขนาดใหญ่และทรานแซคชั่นการผสานกิ่ง
6. ทรานแซคชั่นของการเปลี่ยนแปลงแก้ไขที่นำมาทดสอบในงานวิจัยนี้ คือ ทรานแซคชั่นของการเปลี่ยนแปลงแก้ไขในระดับของแฟ้มข้อมูลและคลาสเท่านั้น ไม่ได้พิจารณาถึงการเปลี่ยนแปลงแก้ไขในระดับเอนทิตี้ที่ละเอียดเช่น เมธอดหรือตัวแปร
7. การทดสอบประสิทธิภาพการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ 3 สถานการณ์แยกจากกัน แต่ใช้ทรานแซคชั่นชุดทดสอบชุดเดียวกันทั้ง 3 สถานการณ์
8. การทดสอบของงานวิจัยนี้เลือกตัวอย่างข้อมูลซอฟต์แวร์อาร์ไคฟว์เพียงตัวอย่างเดียว เนื่องจากงานวิจัยนี้ต้องการวิจัยเพื่อหาข้อมูลเบื้องต้น (Exploratory Research) เท่านั้น

1.7 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้ที่ต้องการพัฒนาระบบให้คำแนะนำสำหรับนักพัฒนาระหว่างการพัฒนาซอฟต์แวร์บนไอดีอี สามารถนำงานวิจัยนี้ไปเป็นแนวทางในการประยุกต์ใช้เข้ากับไอดีอีได้
2. ผู้ที่ต้องการพัฒนาระบบติดตามพัฒนาการการเกิดความเชื่อมโยงกัน (Evolution Coupling) สามารถนำงานวิจัยนี้ไปเป็นแนวทางในการประยุกต์ใช้ได้
3. ผู้ที่ต้องการพัฒนาระบบค้นหารูปแบบการเรียกใช้งานไลบรารี่ (Software Library Call Pattern) สามารถนำงานวิจัยนี้ไปเป็นแนวทางในการประยุกต์ใช้ได้
4. ผลการทดสอบประสิทธิภาพการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ในงานวิจัยนี้ ทำให้ทราบตัวแบบที่เหมาะสำหรับการค้นหากฎความสัมพันธ์ในสถานการณ์การนำทาง (Navigation) สถานการณ์การป้องกันการเกิดข้อผิดพลาด (Error Prevention) และสถานการณ์การเปลี่ยนแปลงแก้ไขที่สมบูรณ์แล้ว (Closure) ในการพัฒนาซอฟต์แวร์ได้
5. ผลการทดสอบประสิทธิภาพการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ในงานวิจัยนี้เป็นประโยชน์สำหรับนักวิจัยที่ต้องการต่อยอดศึกษาการประยุกต์ใช้กฎความสัมพันธ์กับระบบให้คำแนะนำสำหรับนักพัฒนาระหว่างการพัฒนาซอฟต์แวร์ต่อไป

1.8 นิยามศัพท์

1. ซอฟต์แวร์อาร์ไคฟว์ (Software Archives) คือ แฟ้มข้อมูลซอร์สโค้ด (Source Code Files) ทุกเวอร์ชั่น และแฟ้มข้อมูลบันทึกที่ได้จากระบบควบคุมการพัฒนาเวอร์ชั่น (CVS Log File)
2. ซอร์สโค้ด (Source Code) คือ รหัสคอมพิวเตอร์ซึ่งได้รับการเปลี่ยนเป็นภาษาทางเครื่องคอมพิวเตอร์ก่อนทำงานบนเครื่องคอมพิวเตอร์

3. เอนทิตี้ (Entity) คือ เอกลักษณ์หรือสิ่งที่ผู้วิจัยสนใจศึกษา ในที่นี้คำว่า เอนทิตี้ สามารถหมายถึง แฟ้มข้อมูลเอกสาร (File) คลาส (Class) เมธอดหรือฟังก์ชัน (Method or Function) และตัวแปร (Variable)
4. การเปลี่ยนแปลงแก้ไข (Changes) คือ การที่มีนักพัฒนาแก้ไขเอนทิตี้ใดๆ คำว่า การเปลี่ยนแปลงแก้ไขในที่นี้สามารถแสดงได้ 3 มิติ คือ 1) การแก้ไขเอนทิตี้ (alter) 2) การเพิ่มลงในเอนทิตี้ (add to) และ 3) การลบออกจากเอนทิตี้ (delete from)
5. ทรานแซคชั่น (Transaction) สำหรับงานวิจัยนี้ คือ เซตของการเปลี่ยนแปลงแก้ไขที่เกิดขึ้นพร้อมกันหรือในเวลาใกล้เคียงกันและถูกบันทึกเข้าสู่ระบบคอนเคอเรนท์เวอร์ชั่นโดยนักพัฒนาคนเดียวกัน
6. เหตุการณ์ (Situation) คือเซตของการเปลี่ยนแปลงแก้ไขใดๆ ที่เกิดขึ้นจากนักพัฒนา
7. กฎความสัมพันธ์ (Association Rules) สามารถนิยามได้ดังนี้ กำหนดให้ เซต $I = \{i_1, i_2, \ldots, i_m\}$ เป็นเซตของรายการข้อมูล (items) ที่มีอยู่ทั้งหมดและให้ เซต $T = \{t_1, t_2, \ldots, t_n\}$ เป็นเซตของทรานแซคชั่น โดยที่แต่ละทรานแซคชั่น t_n ประกอบด้วยเซตย่อย I_j (j = 1, 2, .. m) ที่เป็นเซตย่อยของเซตของรายการข้อมูล I เซตของรายการข้อมูล I_j นั้นถูกเรียกว่า เซตรายการ (Itemset) ดังนั้นกฎความสัมพันธ์ r ก็คือคู่ของเซตรายการ I_1 และเซตรายการ I_2 โดยที่เซตรายการ I_1 และเซตรายการ I_2 เป็นเซตย่อยของเซต I ที่ไม่มีสมาชิกที่ซ้อนทับกันและเซตรายการ I_2 ไม่เท่ากับเซตว่าง เซตรายการ I_1 ถูกเรียกว่า เซตรายการที่มาก่อน (Antecedent Itemset) และเซตรายการ I_2 ถูกเรียกว่า เซตรายการที่ตามมา (Consequent Itemset) และกำหนดสัญลักษณ์ $I_1 \rightarrow I_2$ แทนกฎความสัมพันธ์ R ที่มีเซตรายการ I_1 เป็นเซตรายการที่มาก่อน และเซตรายการ I_2 เป็นเซตรายการที่ตามมา โดยที่ I_2 ไม่ใช่เซตว่าง (Olivier et al., 2008) สำหรับงานวิจัยนี้ กฎความสัมพันธ์ คือ กฎความสัมพันธ์ที่ตอบข้อถามที่ว่า ถ้านักพัฒนาเปลี่ยนแปลงแก้ไข (เปลี่ยนแปลง เพิ่มลง หรือ ลบออก) เอนทิตี้ใดเอนทิตี้หนึ่งแล้วนักพัฒนาคนนั้นควรจะต้องเปลี่ยนแปลงแก้ไข (เปลี่ยนแปลง เพิ่มลง หรือ ลบออก) เอนทิตี้ใดด้วยต่อไป (Zimmermann et al., 2005; Methanias et al., 2009)

8. คำแนะนำสำหรับเหตุการณ์ Q (Suggestions for Situation Q) คือ เซตของการเปลี่ยนแปลงแก้ไขที่นักพัฒนาควรจะทำตามหลังจากที่นักพัฒนาได้เปลี่ยนแปลงแก้ไขตามเหตุการณ์ Q โดยอ้างอิงมาจากเซตของกฎความสัมพันธ์ที่มีเซตรายการที่มาก่อนเป็นเซตเหตุการณ์ Q (Zimmermann et al., 2005)
9. สถานการณ์การนำทาง (Navigation) คือ สถานการณ์ที่นักพัฒนามีการเปลี่ยนแปลงแก้ไขที่เอนทิตี้หนึ่งแล้ว ระบบจะต้องให้คำแนะนำกับนักพัฒนาให้แก้ไขเอนทิตี้ใดต่อไป (Zimmermann et al., 2005; Methanias et al., 2009)
10. สถานการณ์การป้องกันการเกิดข้อผิดพลาด (Error Prevention) คือ สถานการณ์ที่นักพัฒนามีการเปลี่ยนแปลงแก้ไขที่เอนทิตี้หลายๆเอนทิตี้ต่อเนื่องกันแต่ยังขาดการเปลี่ยนแปลงแก้ไขเอนทิตี้อีกหนึ่งเอนทิตี้จึงจะสมบูรณ์ ระบบจะต้องให้คำแนะนำกับนักพัฒนาให้แก้ไขเอนทิตี้ที่เหลือนั้นได้ (Zimmermann et al., 2005; Methanias et al., 2009)
11. สถานการณ์การเปลี่ยนแปลงแก้ไขที่สมบูรณ์แล้ว (Closure) คือ สถานการณ์ที่นักพัฒนามีการเปลี่ยนแปลงแก้ไขที่เอนทิตี้หลายๆเอนทิตี้ต่อเนื่องกันอย่างครบถ้วนแล้ว ระบบจะต้องไม่ให้คำแนะนำแก้ไขเอนทิตี้ใดๆกับนักพัฒนา (Zimmermann et al., 2005; Methanias et al., 2009)
12. ไอดีอี (IDE: Integrated Development Environment) คือ โปรแกรมประยุกต์ที่จัดเตรียมสิ่งแวดล้อมซึ่งอำนวยความสะดวกให้แก่นักพัฒนาซอฟต์แวร์ โดยปกติแล้วประกอบด้วย เครื่องมือพัฒนาซอร์สโค้ด (Source Code Editor) ตัวแปลภาษาคอมพิวเตอร์ (Compiler) หรือ ตัวแปลคำสั่งคอมพิวเตอร์ (interpreter) หรือทั้งสอง เครื่องมือสร้างระบบอัตโนมัติ (Build Automation Tools) และเครื่องมือตรวจแก้ข้อผิดพลาด (Debugger) เป็นพื้นฐาน