

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การพัฒนาซอฟต์แวร์ในปัจจุบันได้มีความก้าวหน้าไปมาก ไม่ว่าจะเป็นเรื่องของเครื่องมือที่ใช้ในการพัฒนา ความสามารถของนักพัฒนา รวมทั้งผลิตภัณฑ์ซอฟต์แวร์ที่ได้ก็สามารถทำงานได้ตามความต้องการของผู้ใช้ แต่อย่างไรก็ตามผลิตภัณฑ์ซอฟต์แวร์ที่ได้มาอาจจะยังมีข้อบกพร่องอยู่ โดยบางครั้งผู้พัฒนาอาจไม่ทราบได้ว่าผลิตภัณฑ์ซอฟต์แวร์ที่ได้มานั้นมีข้อบกพร่องอะไร และอยู่ที่ตำแหน่งใดในโปรแกรม ทั้งนี้ข้อบกพร่องอาจจะเนื่องมาจากขั้นตอนการออกแบบหรือการเขียนโปรแกรมที่ไม่ดี จึงส่งผลให้การแก้ไขปรับปรุงซอฟต์แวร์และการทำความเข้าใจในซอฟต์แวร์ในอนาคตทำได้ยากขึ้น

นักวิจัยได้พยายามค้นหาปัญหาในการพัฒนาซอฟต์แวร์ และนำเสนอวิธีการแก้ปัญหาเหล่านั้นมาโดยตลอด วิธีการหนึ่งที่ถูกนำเสนอ คือ การพิจารณาจากปัญหาที่พบบ่อยๆในการพัฒนาซอฟต์แวร์ และได้รวบรวมปัญหาที่พบเหล่านี้ไว้ เรียกว่า ร่องรอยที่ไม่ดี (Bad-smell) ร่องรอยที่ไม่ดี คือ ลักษณะของการออกแบบซอฟต์แวร์และการเขียนโปรแกรมที่ไม่ดี ทำให้การทำความเข้าใจซอฟต์แวร์ และการแก้ไขซอฟต์แวร์ทำได้ยากขึ้น

การค้นหาร่องรอยที่ไม่ดีเป็นขั้นตอนที่สำคัญก่อนที่จะถึงขั้นตอนการทำ รีแฟคตอริง (Refactoring) ดังนั้นถ้าไม่มีวิธีการ และ เครื่องมือที่ช่วยในการค้นหาร่องรอยที่ไม่ดีที่มีอยู่ในโปรแกรมจะทำให้เกิดปัญหาตามมา เช่น ทำให้เสียเวลา และ กำลังคนในการทำขั้นตอนนี้มาก เพราะผู้พัฒนาอาจขาดความรู้ในเรื่องการค้นหาร่องรอยที่ไม่ดี และ แต่ละโปรแกรมจะมีร่องรอยที่ไม่ดีที่แตกต่างกัน รวมทั้งโปรแกรมขนาดใหญ่มีจำนวนความยาวของบรรทัดหลายพันบรรทัด เป็นต้น ถ้าไม่มีเครื่องมือที่ช่วยในการค้นหาจะทำให้เสียเวลามาก จึงมีความจำเป็นอย่างยิ่งที่ต้องมีเครื่องมือมาช่วยในการค้นหาร่องรอยที่ไม่ดีเหล่านั้น

ดังนั้นผู้วิจัยจึงมีวัตถุประสงค์ที่จะออกแบบ และ พัฒนาเครื่องมือที่ช่วยในการค้นหาร่องรอยที่ไม่ดีสำหรับโปรแกรมที่พัฒนาด้วยภาษาจาวา โดยจะออกแบบวิธีการค้นหา และ พัฒนาเครื่องมือที่ใช้ในการค้นหาร่องรอยที่ไม่ดีโดยการพิจารณาจากโครงสร้างของโปรแกรม ในงานวิจัยนี้จะนำเสนอวิธีการค้นหาร่องรอยที่ไม่ดี 4 ประเภทด้วยกัน คือ Feature Envy, Message Chains, Middle Man, และ Inappropriate Intimacy (General Form)

1.2 วัตถุประสงค์

เพื่อออกแบบ และ พัฒนาเครื่องมือค้นหาร่องรอยที่ไม่ดีสำหรับโปรแกรมเชิงวัตถุที่พัฒนาด้วยภาษาจาวา

1.3 ขอบเขตของงานวิจัย

1. ออกแบบเพรดิเคตสำหรับโปรแกรมเชิงวัตถุ
2. ออกแบบกฎโปรล็อกสำหรับร่องรอยที่ไม่ดีอย่างน้อย 4 ประเภท
3. พัฒนาเครื่องมือค้นหาร่องรอยที่ไม่ดีโดยใช้หลักการของโปรล็อก
4. การประเมินความสามารถของเครื่องมือ
 - 4.1 ซอฟต์แวร์ที่นำมาค้นหาร่องรอยที่ไม่ดีต้องพัฒนาด้วยภาษาจาวา และ ไม่มีข้อผิดพลาดทางไวยากรณ์
 - 4.2 จำนวนซอฟต์แวร์ที่ใช้ในการทดสอบกับเครื่องมืออย่างน้อย 3 ระบบ
 - 4.3 ขนาดของซอฟต์แวร์ที่ใช้ในการทดสอบต้องประกอบไปด้วยอย่างน้อย 5 คลาสขึ้นไป
5. พัฒนาเครื่องมือ และ ใช้เครื่องมือบนระบบปฏิบัติการวินโดวส์ (Windows) ตั้งแต่รุ่น 98 ขึ้นไป

1.4 ขั้นตอนและวิธีดำเนินงานวิจัย

1. ศึกษาแนวคิด และ ทฤษฎีที่เกี่ยวข้องกับร่องรอยที่ไม่ดี
2. ศึกษาทฤษฎีที่เกี่ยวกับตรรกะเพรดิเคต และ โปรล็อก
3. ศึกษาเครื่องมือที่ช่วยในการสร้างตัวแปลภาษา ที่มีการสร้างซอร์สโค้ดเป็นภาษาจาวา ได้แก่ โปรแกรม จาวาซีซี (Javacc) ของบริษัทซันไมโครซิสเต็มส์ (Sun Microsystem, Inc.)
4. ศึกษาหลักไวยากรณ์ต่างๆ ของโปรแกรมภาษาจาวาเพื่อหาวิธีในการค้นหาร่องรอยที่ไม่ดี
5. เลือกร่องรอยที่ไม่ดี
6. ออกแบบวิธีการค้นหาร่องรอยที่ไม่ดี
7. ออกแบบ และ พัฒนาเครื่องมือที่ใช้ในการค้นหาร่องรอยที่ไม่ดีในโปรแกรมที่พัฒนาด้วยภาษาจาวา
8. ทดสอบเครื่องมือค้นหาร่องรอยที่ไม่ดีในโปรแกรมที่พัฒนาด้วยภาษาจาวา
9. สรุปผลการวิจัย และ ข้อเสนอแนะ

10. จัดทำรายงานวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำเครื่องมือคั้นหว่ารงรอยที่ไม่ดีไปใช้เป็นแนวทางเพื่อปรับปรุงซอร์สโค้ดได้
2. ได้เพรดิเคตที่แสดงความสัมพันธ์ขององค์ประกอบต่างๆ ในโปรแกรม เช่น คลาส เมธอด การเรียกใช้เมธอด เป็นต้น ที่ใช้สร้างข้อเท็จจริง เพื่อคั้นหว่ารงรอยที่ไม่ดี
3. ได้กฎโปรล็อกของรื่องรอยที่ไม่ดี ทั้ง 4 ประเภท เพื่อใช้ในการคั้นหว่ารงรอยที่ไม่ดี