การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ในเอ็กซ์เอดีแอล

นายอาทิตย์ อุดมสมฤดี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR)
are the thesis authors' files submitted through the Graduate School.

EFFECT DETECTION OF SOFTWARE ARCHITECTURE CHANGES IN xADL

Mr. Artit Udomsomruedee

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2012

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ในเอ็กซ์เอดีแอล
โดย	นายอาทิตย์ อุดมสมฤดี
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ

คณะวิศกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

…….. คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

…… ประธานกรรมการ
(รองศาสตราจารย์ ดร.ธาราทิพย์ สุวรรณศาสตร์)

……... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)

…… กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.อาทิตย์ ทองทักษ์)

…… กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.ทรงศักดิ์ รองวิริยะพานิช)

อาทิตย์ อุดมสมฤดี : การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ในเอ็กซ์เอดีแอล. (EFFECT DETECTION OF SOFTWARE ARCHITECTURE CHANGES IN xADL) อ. ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร.วิวัฒน์ วัฒนาวุฒิ, 169 หน้า.

สถาปัตยกรรมซอฟต์แวร์เป็นผลจากกระบวนการในการกำหนดแนวทางการแก้ไขปัญหาจากความต้องการทางเทคนิคและความต้องการในการปฏิบัติงาน โดยสถาปัตยกรรมซอฟต์แวร์เกี่ยวข้องกับส่วนประกอบต่างๆ รวมถึงความสัมพันธ์กันระหว่างส่วนประกอบ ซึ่งถูกสร้างขึ้นก่อนขั้นตอนการพัฒนาซอฟต์แวร์ การออกแบบสถาปัตยกรรมซอฟต์แวร์ที่ดีมีผลโดยตรงต่อคุณภาพของผลิตภัณฑ์ซอฟต์แวร์ที่พัฒนาขึ้น การเปลี่ยนแปลงแบบจำลองสถาปัตยกรรมซอฟต์แวร์ในระหว่างช่วงของการพัฒนาซอฟต์แวร์สามารถเกิดขึ้นได้ ซึ่งอาจทำให้เกิดผลกระทบต่อเหตุผลในการออกแบบ ประสิทธิภาพ และความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์

งานวิจัยนี้ นำเสนอการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ สนับสนุนการทำงานโดยเครื่องมือที่พัฒนาขึ้น เรียกว่า “xADL: Software Architecture Changes Effect Detection Tools – xSACEDT” เพื่อใช้ในการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ในภาษาเอ็กซ์เอดีแอลที่ได้จากเครื่องมือ Archstudio 4 โดยเปรียบเทียบกับแบบจำลองที่ถูกแก้ไข การเปลี่ยนแปลงที่เกิดขึ้นสามารถถูกตรวจจับและแสดงผลลัพธ์ออกมาในรูปแบบรายงาน นอกจากนี้ ยังสามารถติดตามผลกระทบที่เกิดขึ้นกับรายการข้อกำหนดความต้องการและคุณลักษณะเชิงคุณภาพและนำมาสรุปเป็นผลกระทบที่เกิดจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ได้

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2555 .....

5471037721 : MAJOR SOFTWARE ENGINEERING

KEYWORDS : EFFECT DETECTION / ARCHITECTURE MODEL / QUALITY ATTRIBUTE / SOFTWARE ARCHITECTURE / DESIGN RATIONALE

ARTIT UDOMSOMRUEDEE : EFFECT DETECTION OF SOFTWARE ARCHITECTURE CHANGES IN xADL. ADVISOR : ASST.PROF. WIWAT VATANAWOOD, Ph.D., 169 pp.

Software architecture is an output of defining solution process that meets all of the technical and operational requirements. It involves a set of related software elements and their relationships to be constructed afterwards in the rest of development phases. A good software architecture design obviously ensures the quality of software product. Typically, the changes of software architecture model during the development phases may effect the expected design rationales, its performance and the complexity of the software product.

In this paper, we propose an effects detection of software architecture changes that supporting by our specific development tools called "xADL: Software Architecture Changes Effect Detection Tools - xSACEDT". The original software architecture model, written in xADL by Archstudio 4 development tools, will be compared with the modified model. All of the modification issues will be detected and reported. Moreover, the effects on requirement checklists and its design quality attributes are also traced and alerted.

Department : Computer Engineering Student's Signature .....

Field of Study : Software Engineering Advisor's Signature .....

Academic Year : 2012

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้จะไม่สามารถทำได้อย่างเสร็จลุล่วงด้วยดีได้เลยถ้าขาดความช่วยเหลือ คำแนะนำ ตลอดจนข้อคิดเห็นที่ได้รับจาก รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ช่วยสละเวลาอันมีค่าและคอยให้เวลาในการมอบความช่วยเหลือในทุกด้านเป็นอย่างมาก อีกทั้ง ยังคอยชี้แนะแนวทางในการดำเนินงานเกี่ยวกับงานวิจัยโดยรวม การติดตามกำกับดูแลและช่วยแก้ไขปัญหาต่างๆในระหว่างดำเนินการวิจัยจนทำให้ผู้ทำวิจัยสามารถทำงานวิจัยจนประสบผลสำเร็จได้

ขอขอบพระคุณ รองศาสตราจารย์ ดร.ธาราทิพย์ สุวรรณศาสตร์ ประธานกรรมการสอบ ผู้ช่วยศาสตราจารย์ ดร.อาทิตย์ ทองทักษ์ และผู้ช่วยศาสตราจารย์ ดร.ทรงศักดิ์ รองวิริยะพานิช กรรมการสอบวิทยานิพนธ์ ที่ได้กรุณาเสียสละเวลาอันมีค่าอย่างยิ่ง เพื่อให้คำแนะนำ แนวทางในการทำงานวิจัยให้วิทยานิพนธ์ฉบับนี้ได้สำเร็จเสร็จสิ้นโดยเป็นไปอย่างมีคุณภาพ

ขอขอบพระคุณคณะคณาจารย์ทุกท่าน ที่ให้ความรู้ สั่งสอนแนวทางในการดำเนินชีวิต และสั่งสอนความอดทน ความพากเพียร ความมานะ ทุ่มเทในการทำงาน ตลอดจนความห่วงใยในการทำวิจัย

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ สาขาวิชาวิศวกรรมซอฟต์แวร์ จุฬาลงกรณ์มหาวิทยาลัย ทุกคนที่คอยให้กำลังใจ ความคิดเห็น คำแนะนำในการทำงานวิจัยให้บรรลุผลสำเร็จ

สิ่งสำคัญที่สุด ต้องกราบขอบพระคุณบิดา มารดา และสมาชิกในครอบครัวทุกคนที่เป็นกำลังใจสำคัญ เอาใจใส่ดูแล และให้การสนับสนุนในการศึกษาของผู้ทำวิจัยเรื่อยมา

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ง

บทคัดย่อภาษาอังกฤษ..... จ

กิตติกรรมประกาศ..... ฉ

สารบัญ..... ช

สารบัญตาราง..... ญ

สารบัญรูป..... ฌ

บทที่ 1 บทนำ..... 1

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของงานวิจัย..... 3

1.3 ขอบเขตของการวิจัย..... 3

1.4 ลำดับขั้นตอนในการนำเสนองานวิจัย..... 5

1.5 บทความวิชาการที่ได้รับการตีพิมพ์..... 6

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง..... 7

2.1 ทฤษฎีที่เกี่ยวข้อง..... 7

2.1.1 สถาปัตยกรรมซอฟต์แวร์ (Software Architecture)..... 7

2.1.2 ภาษาเอดีแอล (ADL - Architecture description language)..... 8

2.1.3 ภาษาเอ็กซ์เอดีแอล (xADL - Highly-Extension architecture description language)..... 8

2.1.4 เอ็กซ์เอ็มแอล, ดีทีดี และ เค้าร่างเอ็กซ์เอ็มแอล (XML, DTDs and XML Schemas)..... 9

2.1.5 คุณลักษณะเชิงคุณภาพของซอฟต์แวร์ (Software Quality Attributes)..... 9

2.1.6 ระบบซีวีเอส (CVS - Concurrent Versions System)..... 10

2.1.7 ตารางแฮช (Hash Table)..... 11

2.2 งานวิจัยที่เกี่ยวข้อง..... 12

2.2.1 Composing Structural Views in xADL..... 12

2.2.2 A Highly-Extensible, XML-Based Architecture Description Language..... 13

หน้า

2.2.3 A rationale-based architecture model for design traceability and..... reasoning..... 17
บทที่ 3 การวิเคราะห์และการออกแบบการตรวจจับผลกระทบของการเปลี่ยนแปลง..... สถาปัตยกรรมซอฟต์แวร์..... 20
3.1 กำหนดส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ที่สนใจในการตรวจจับการ..... เปลี่ยนแปลง..... 20
3.2 การจัดเก็บคำอธิบายข้อกำหนดความต้องการโครงการและการเชื่อมโยงกับ..... คำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์..... 22
3.3 ออกแบบวิธีการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรม..... ซอฟต์แวร์..... 24
3.4 ผลกระทบที่ได้จากการตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์..... 29
บทที่ 4 การออกแบบและการพัฒนาเครื่องมือ..... 31
4.1 โครงสร้างเครื่องมือการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรม..... ซอฟต์แวร์ในเอ็กซ์เอดีแอล..... 31
4.2 พัฒนาเครื่องมือที่ใช้ในการตรวจจับผลกระทบจากการเปลี่ยนแปลง..... สถาปัตยกรรมซอฟต์แวร์..... 33
4.2.1 แผนภาพยูสเคส..... 34
4.2.2 แผนภาพคลาส..... 40
4.2.3 แผนภาพกิจกรรม..... 44
4.3 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ..... 47
4.4 โครงสร้างส่วนต่อประสานกับผู้ใช้เครื่องมือ..... 47
4.4.1 ส่วนการจัดการคำอธิบายข้อกำหนดความต้องการโครงการ..... 48
4.4.2 ส่วนการจัดการคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรม..... ซอฟต์แวร์..... 52
4.4.3 ส่วนการตรวจจับความเปลี่ยนแปลงและสรุปผลกระทบจากการ..... เปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์..... 55
บทที่ 5 การทดสอบเครื่องมือ..... 59
5.1 การทดสอบเครื่องมือ..... 59

หน้า

5.1.1 กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี (กำหนดเงื่อนไขการ.....
เปลี่ยนแปลงให้ครอบคลุมกับขอบเขตความสามารถในการตรวจจับของ.....
เครื่องมือ)..... 60
5.1.2 กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี..... 73
5.1.3 กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์..... 86
5.1.4 กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย..... 96
บทที่ 6 สรุปผลการวิจัย..... 111
6.1 สรุปผลการวิจัย..... 111
6.2 ประโยชน์ที่ได้รับจากงานวิจัย..... 111
6.3 ข้อจำกัดของเครื่องมือ..... 112
6.4 แนวทางในการพัฒนางานวิจัยต่อ..... 112
รายการอ้างอิง..... 113
ภาคผนวก..... 116
ภาคผนวก ก คำอธิบายโครงการและคำอธิบายข้อกำหนดความต้องการโครงการ.....
ระบบที่ใช้ในการทดสอบ..... 117
ภาคผนวก ข คำอธิบายการออกแบบสถาปัตยกรรมซอฟต์แวร์ระบบที่ใช้ในการ.....
ทดสอบ..... 142
ภาคผนวก ค ผลการทดสอบการตรวจจับผลกระทบของการเปลี่ยนแปลง.....
สถาปัตยกรรมซอฟต์แวร์..... 150
ประวัติผู้เขียนวิทยานิพนธ์..... 169

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงโครงสร้างของภาษาเอ็กซ์เอดีแอล 2.0 แสดงตามจุดประสงค์การ…..…. ออกแบบ [6]…….....… 16
ตารางที่ 4.1 คำอธิบายยูสเคสจัดการคำอธิบายข้อกำหนดความต้องการโครงการ.....…. 35
ตารางที่ 4.2 คำอธิบายยูสเคสนำเข้าคำอธิบายข้อกำหนดความต้องการโครงการ.....…. 35
ตารางที่ 4.3 คำอธิบายยูสเคสส่งออกคำอธิบายข้อกำหนดความต้องการโครงการ.....…. 36
ตารางที่ 4.4 คำอธิบายยูสเคสจัดการคำอธิบายการออกแบบส่วนประกอบ……..…. สถาปัตยกรรมซอฟต์แวร์…..……..……..…. 36
ตารางที่ 4.5 คำอธิบายยูสเคสนำเข้าแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์.....…. 37
ตารางที่ 4.6 คำอธิบายยูสเคสตรวจจับความเปลี่ยนแปลงและสรุปผลกระทบจากการ…..... เปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์……..... 38
ตารางที่ 4.7 คำอธิบายยูสเคสนำเข้าสถาปัตยกรรมซอฟต์แวร์ตั้งฉบับและสถาปัตยกรรม..... ซอฟต์แวร์ที่ถูกแก้ไข……..……..….. 39
ตารางที่ 4.8 รายละเอียดตารางความสัมพันธ์ระหว่างคุณลักษณะกับแผนภาพคลาส…..…. จาวาอ็อบเจ็กต์……..……..…. 41
ตารางที่ 4.9 โครงสร้างข้อมูลของคลาส Object……..……..…… 41
ตารางที่ 4.10 โครงสร้างข้อมูลของคลาส Pointer……..……..…… 41
ตารางที่ 4.11 โครงสร้างข้อมูลของคลาส Link……..…… 42
ตารางที่ 4.12 โครงสร้างข้อมูลของคลาส Connector……..…… 42
ตารางที่ 4.13 โครงสร้างข้อมูลของคลาส Component ……..…… 42
ตารางที่ 4.14 โครงสร้างข้อมูลของคลาส Interface…..……..…… 42
ตารางที่ 4.15 โครงสร้างข้อมูลของคลาส Type……..……..…… 43
ตารางที่ 4.16 โครงสร้างข้อมูลของคลาส Structure…..……..…… 43
ตารางที่ 4.17 โครงสร้างข้อมูลของคลาส Requirement……..…… 43
ตารางที่ 4.18 โครงสร้างข้อมูลของคลาส Project……..…… 44
ตารางที่ 5.1 ตัวอย่างและผลลัพธ์การบันทึกคำอธิบายโครงการ กรณีทดสอบที่ 1……..…. ระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M02…..……..…… 64

หน้า

ตารางที่ 5.2 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ……..……..
กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ที่ 9.....……..…….....….. 65
ตารางที่ 5.3 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ……
กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ที่ 10……..……...…… 65
ตารางที่ 5.4 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ……..……
กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ที่ 11…….....……..……..... 66
ตารางที่ 5.5 ตัวอย่างและผลการบันทึกคำอธิบายการออกแบบส่วนประกอบของ…….....
สถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี…....…
รหัสโครงการ ABC-CODE-M02…….……..……..……..……..... 67
ตารางที่ 5.6 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรม.....
ซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี……..……
ส่วนผลกระทบต่อความต้องการซอฟต์แวร์.....……..……..…….....….. 69
ตารางที่ 5.7 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรม.....
ซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี……..…
ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ……..……..……..…… 70
ตารางที่ 5.8 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรม.....
ซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี……
ส่วนผลกระทบต่อชุดคำสั่ง.....……..... 70
ตารางที่ 5.9 ตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์…....
กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี.....……..……..... 71
ตารางที่ 5.10 ตัวอย่างการตรวจจับรายละเอียดการเปลี่ยนแปลงส่วนประกอบ……....
กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี.....……...……..... 72
ตารางที่ 5.11 ตัวอย่างและผลลัพธ์การบันทึกคำอธิบายโครงการ กรณีทดสอบที่ 2……..…
ระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M02……..... 77
ตารางที่ 5.12 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ……..…
กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ที่ 9……..……..…….....….. 78
ตารางที่ 5.13 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ……
กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ที่ 10…….....……..…….. 78

หน้า

ตารางที่ 5.14 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ……
กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ที่ 11…..……..…..…….. 79
ตารางที่ 5.15 ตัวอย่างและผลการบันทึกคำอธิบายการออกแบบส่วนประกอบของ……..
สถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี..……
รหัสโครงการ ABC-CODE-M02……..……..……..……..…..…. 81
ตารางที่ 5.16 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรม.....
ซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี……
ส่วนผลกระทบต่อความต้องการซอฟต์แวร์.....……..……..…..…. 82
ตารางที่ 5.17 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรม.....
ซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี……
ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ……..……..……..…….. 83
ตารางที่ 5.18 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรม.....
ซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี……
ส่วนผลกระทบต่อชุดคำสั่ง.....…….....…….....…….

หน้า

ตารางที่ 5.25 ตัวอย่างและผลการบันทึกคำอธิบายการออกแบบส่วนประกอบของ…..
สถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 3……..
ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ รหัสโครงการ.....
ARCADE-CODE-A01……..... 92
ตารางที่ 5.26 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรม.....
ซอฟต์แวร์ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์……
ส่วนผลกระทบต่อความต้องการซอฟต์แวร์.....……..……..... 93
ตารางที่ 5.27 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรม.....
ซอฟต์แวร์ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์……...
ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ……..……..……. 94
ตารางที่ 5.28 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรม.....
ซอฟต์แวร์ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์……..…
ส่วนผลกระทบต่อชุดคำสั่ง..……..... 95
ตารางที่ 5.29 ตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์…....
กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์……..…..... 95
ตารางที่ 5.30 ตัวอย่างการตรวจจับรายละเอียดการเปลี่ยนแปลงส่วนประกอบ……...
กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์..……..…..... 96
ตารางที่ 5.31 ตัวอย่างและผลลัพธ์การบันทึกคำอธิบายโครงการ กรณีทดสอบที่ 3……..….
ระบบการลงทะเบียนมหาวิทยาลัย รหัสโครงการ UNI-CODE-2013……..... 101
ตารางที่ 5.32 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ……..……
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ที่ 1……..…..... 102
ตารางที่ 5.33 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ……..……
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ที่ 2……..…..... 102
ตารางที่ 5.34 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ……..……
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ที่ 3……..…..... 103
ตารางที่ 5.35 ตัวอย่างและผลการบันทึกคำอธิบายการออกแบบส่วนประกอบของ…..
สถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 4 ……..……
ระบบการลงทะเบียนมหาวิทยาลัย รหัสโครงการ UNI-CODE-2013……..... 104

หน้า

ตารางที่ 5.36 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรม.....
ซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย……..…….....…
ส่วนผลกระทบต่อความต้องการซอฟต์แวร์.....……..……..……..... 106
ตารางที่ 5.37 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรม.....
ซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย……..……
ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ……..……..……..…….. 106
ตารางที่ 5.38 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรม.....
ซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย…...…….....…
ส่วนผลกระทบต่อชุดคำสั่ง..……..... 108
ตารางที่ 5.39 ตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์.....…
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย……..……..... 108
ตารางที่ 5.40 ตัวอย่างการตรวจจับรายละเอียดการเปลี่ยนแปลงส่วนประกอบ……..……
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย……..……..……..…… 109
ตารางที่ ก-1 คำอธิบายโครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ……..…
ABC-CODE-M01…..……..……..……..…….. 117
ตารางที่ ก-2 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 1..…….... 118
ตารางที่ ก-3 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 2..…….... 118
ตารางที่ ก-4 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 3..…….... 118
ตารางที่ ก-5 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 4..…….... 119
ตารางที่ ก-6 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 5..…….... 120
ตารางที่ ก-7 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 6..…….... 121
ตารางที่ ก-8 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 7..…….... 122
ตารางที่ ก-9 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 8..…….... 123
ตารางที่ ก-10 คำอธิบายโครงการระบบบริหารงานบริษัทเอบีซีรหัสโครงการ……...
ABC-CODE-M02…..……..……..……..…….. 123
ตารางที่ ก-11 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 1..…….... 124
ตารางที่ ก-12 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 2..…….... 125
ตารางที่ ก-13 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 3..…….... 125
ตารางที่ ก-14 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 4..…….... 126

หน้า

ตารางที่ ก-15 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 5..... 127
ตารางที่ ก-16 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 6..... 127
ตารางที่ ก-17 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 7..... 128
ตารางที่ ก-18 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 8..... 129
ตารางที่ ก-19 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 9..... 130
ตารางที่ ก-20 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 10..... 130
ตารางที่ ก-21 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 11..... 131
ตารางที่ ก-22 คำอธิบายโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์..... 132
ตารางที่ ก-23 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์.....
ที่ 1..... 132
ตารางที่ ก-24 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์.....
ที่ 2..... 133
ตารางที่ ก-25 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์.....
ที่ 3..... 134
ตารางที่ ก-26 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์.....
ที่ 4..... 134
ตารางที่ ก-27 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์.....
ที่ 5..... 135
ตารางที่ ก-28 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์.....
ที่ 6..... 135
ตารางที่ ก-29 คำอธิบายโครงการระบบการลงทะเบียนมหาวิทยาลัย..... 136
ตารางที่ ก-30 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย.....
ที่ 1..... 136
ตารางที่ ก-31 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย.....
ที่ 2..... 137
ตารางที่ ก-32 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย.....
ที่ 3..... 138
ตารางที่ ก-33 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย.....
ที่ 4..... 138

หน้า

ตารางที่ ก-34 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย…….. ที่ 5..... 139
ตารางที่ ก-35 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย…….. ที่ 6..... 139
ตารางที่ ก-36 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย…….. ที่ 7..... 140
ตารางที่ ก-37 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย…….. ที่ 8..... 141
ตารางที่ ข-1 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์……..... กรณีทดสอบที่ 1 โครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ……..... ABC-CODE-M01..... 142
ตารางที่ ข-2 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์……..... กรณีทดสอบที่ 1 โครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ……..... ABC-CODE-M02..... 143
ตารางที่ ข-3 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์……..... กรณีทดสอบที่ 2 โครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ……..... ABC-CODE-M01..... 144
ตารางที่ ข-4 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์……..... กรณีทดสอบที่ 2 โครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ……..... ABC-CODE-M02..... 145
ตารางที่ ข-5 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์……..... กรณีทดสอบที่ 3 โครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์……..... รหัสโครงการ ARCADE-CODE-A01..... 147
ตารางที่ ข-6 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์……..... กรณีทดสอบที่ 4 โครงการระบบการลงทะเบียนมหาวิทยาลัย……..... รหัสโครงการ UNI-CODE-2013..... 148
ตารางที่ ค-1 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์... กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อความ……..... ต้องการซอฟต์แวร์..... 150

หน้า

ตารางที่ ค-2 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์...
กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อคุณลักษณะ..
เชิงคุณภาพ..... 151
ตารางที่ ค-3 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์...
กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อชุดคำสั่ง..... 151
ตารางที่ ค-4 ผลลัพธ์ความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์.....
กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี..... 152
ตารางที่ ค-5 ผลลัพธ์รายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์.....
กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี..... 152
ตารางที่ ค-6 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์...
กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อความ.....
ต้องการซอฟต์แวร์..... 153
ตารางที่ ค-7 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์...
กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อคุณลักษณะ..
เชิงคุณภาพ..... 154
ตารางที่ ค-8 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์...
กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อชุดคำสั่ง..... 155
ตารางที่ ค-9 ผลลัพธ์ความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์.....
กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี..... 156
ตารางที่ ค-10 ผลลัพธ์รายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์.....
กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี..... 158
ตารางที่ ค-11 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์...
กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์.....
ส่วนผลกระทบต่อความต้องการซอฟต์แวร์..... 159
ตารางที่ ค-12 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์...
กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์.....
ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ..... 159

หน้า

ตารางที่ ค-13 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์...
กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์……..……..
ส่วนผลกระทบต่อชุดคำสั่ง……...…... 160
ตารางที่ ค-14 ผลลัพธ์ความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์…….....
กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์..... 161
ตารางที่ ค-15 ผลลัพธ์รายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์……..……
กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์..... 162
ตารางที่ ค-16 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์...
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย……..…..
ส่วนผลกระทบต่อความต้องการซอฟต์แวร์..... 162
ตารางที่ ค-17 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์...
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย……..…..
ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ..... 163
ตารางที่ ค-18 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์...
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย……..…..
ส่วนผลกระทบต่อชุดคำสั่ง..... 165
ตารางที่ ค-19 ผลลัพธ์ความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์…….....
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย..... 165
ตารางที่ ค-20 ผลลัพธ์รายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์……..……
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย..... 167

สารบัญรูป

หน้า

รูปที่ 1.1 ภาพรวมการออกแบบสถาปัตยกรรมซอฟต์แวร์ [3]..... 2
รูปที่ 2.1 ตัวอย่างการจัดเก็บโครงสร้างข้อมูลแบบตารางแฮช..... 11
รูปที่ 2.2 รายละเอียดส่วนประกอบของภาษาเอ็กซ์เอดีแอล [18]..... 12
รูปที่ 3.1 ตัวอย่างแถบข้อมูลส่วนประกอบ..... 21
รูปที่ 3.2 ตัวอย่างแถบข้อมูลตัวเชื่อม..... 21
รูปที่ 3.3 ตัวอย่างแถบข้อมูลจุดเชื่อมต่อ..... 21
รูปที่ 3.4 ตัวอย่างแถบข้อมูลเส้นเชื่อม..... 22
รูปที่ 3.5 ตัวอย่างรูปแบบแฟ้มข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ..... 23
รูปที่ 3.6 ตัวอย่างการจัดเก็บคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรม.....
ซอฟต์แวร์..... 24
รูปที่ 3.7 การอ่านโครงสร้างสถาปัตยกรรมซอฟต์แวร์ต้นฉบับและสถาปัตยกรรม.....
ซอฟต์แวร์ที่ถูกแก้ไข และการตรวจจับความเปลี่ยนแปลง..... 25
รูปที่ 3.8 การตรวจจับผลกระทบ..... 27
รูปที่ 3.9 การแสดงผลรายงาน..... 28
รูปที่ 3.10 ตัวอย่างแถบข้อมูลคำอธิบายชุดคำสั่งที่เกี่ยวข้องกับประเภทส่วนประกอบ... 30
รูปที่ 4.1 ภาพรวมโครงสร้างเครื่องมือการตรวจจับผลกระทบของการเปลี่ยนแปลง.....
สถาปัตยกรรมซอฟต์แวร์ในเอ็กซ์เอดีแอล..... 31
รูปที่ 4.2 แผนภาพยูสเคสของเครื่องมือที่ใช้สำหรับการตรวจจับผลกระทบจากการ.....
เปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์..... 34
รูปที่ 4.3 แผนภาพคลาสของเครื่องมือที่ใช้ในการตรวจจับผลกระทบจากการ.....
เปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์..... 40
รูปที่ 4.4 แผนภาพกิจกรรมการจัดการคำอธิบายข้อกำหนดความต้องการโครงการ..... 44
รูปที่ 4.5 แผนภาพกิจกรรมการจัดการคำอธิบายการออกแบบส่วนประกอบ.....
สถาปัตยกรรมซอฟต์แวร์..... 45
รูปที่ 4.6 แผนภาพกิจกรรมการตรวจจับความเปลี่ยนแปลงและสรุปผลกระทบ.....
จากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์..... 46
รูปที่ 4.7 แผนภาพส่วนประกอบของเครื่องมือ..... 48

หน้า

รูปที่ 4.8 ตัวอย่างส่วนต่อประสานหลักกับผู้ใช้งานสำหรับการจัดการข้อมูลคำอธิบาย..
ข้อกำหนดความต้องการโครงการ..... 49
รูปที่ 4.9 ตัวอย่างส่วนต่อประสานกับผู้ใช้งานแสดงผลรายการข้อกำหนด.....
ความต้องการที่เกี่ยวกับโครงการซอฟต์แวร์.....50
รูปที่ 4.10 ตัวอย่างส่วนต่อประสานกับผู้ใช้งานสำหรับการบันทึกข้อมูลโครงการ.....
เบื้องต้น..... 51
รูปที่ 4.11 ตัวอย่างส่วนต่อประสานกับผู้ใช้งานสำหรับการบันทึกข้อมูลข้อกำหนด.....
ความต้องการของโครงการ..... 51
รูปที่ 4.12 ตัวอย่างส่วนต่อประสานหลักกับผู้ใช้งานสำหรับจัดการคำอธิบาย.....
การออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์..... 52
รูปที่ 4.13 ตัวอย่างส่วนต่อประสานกับผู้ใช้งานแสดงส่วนประกอบสถาปัตยกรรม.....
ซอฟต์แวร์หลังจากนำเข้าแฟ้มข้อมูล..... 53
รูปที่ 4.14 ตัวอย่างส่วนต่อประสานกับผู้ใช้งานสำหรับการกำหนดคำอธิบาย.....
การออกแบบให้กับส่วนประกอบ..... 54
รูปที่ 4.15 ตัวอย่างส่วนต่อประสานกับผู้ใช้งานสำหรับการตรวจจับความเปลี่ยนแปลง.
และสรุปผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์. 55
รูปที่ 4.16 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงโครงสร้าง.....
สถาปัตยกรรมซอฟต์แวร์ในรูปแบบรายงาน..... 56
รูปที่ 4.17 ตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงโครงสร้างสถาปัตยกรรม....
ซอฟต์แวร์ในรูปแบบรายงาน..... 57
รูปที่ 4.18 ตัวอย่างผลลัพธ์รายละเอียดการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรม.....
ซอฟต์แวร์ในรูปแบบรายงาน..... 58
รูปที่ 5.1 กรณีทดสอบที่ 1 ภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงาน.....
บริษัทเอบีซี ก่อนการเปลี่ยนแปลง..... 61
รูปที่ 5.2 กรณีทดสอบที่ 1 ภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงาน.....
บริษัทเอบีซี หลังการเปลี่ยนแปลง..... 63
รูปที่ 5.3 กรณีทดสอบที่ 2 ภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงาน.....
บริษัทเอบีซี ก่อนการเปลี่ยนแปลง..... 74

หน้า

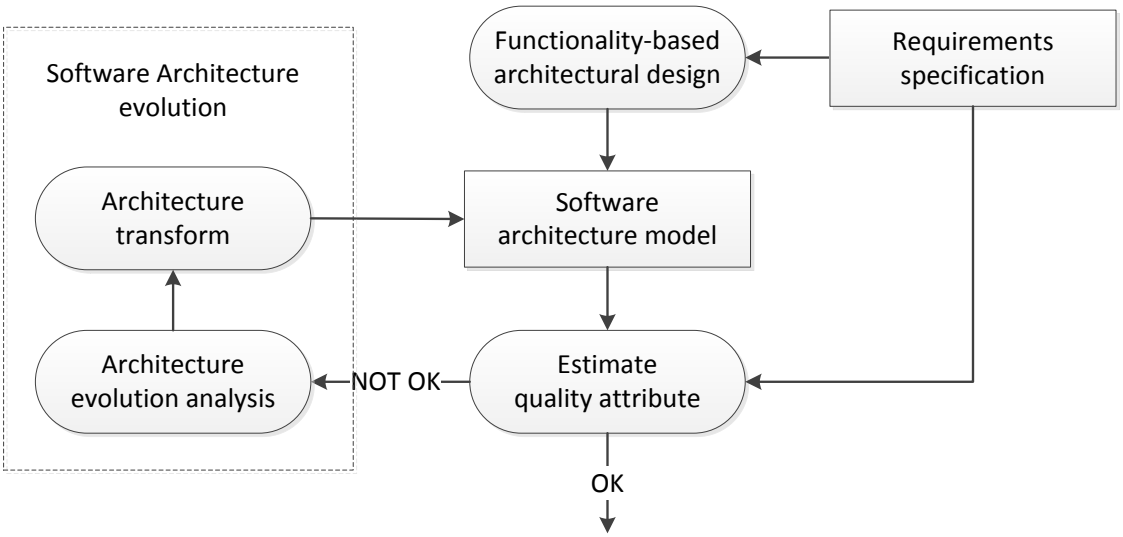
รูปที่ 5.4 กรณีทดสอบที่ 2 ภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงาน……..
บริษัทเอบีซี หลังการเปลี่ยนแปลง……..... 76
รูปที่ 5.5 กรณีทดสอบที่ 3 ภาพรวมสถาปัตยกรรมซอฟต์แวร์……..
ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ก่อนการเปลี่ยนแปลง……..... 87
รูปที่ 5.6 กรณีทดสอบที่ 3 ภาพรวมสถาปัตยกรรมซอฟต์แวร์…….....
ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ หลังการเปลี่ยนแปลง……..... 88
รูปที่ 5.7 กรณีทดสอบที่ 4 ภาพรวมสถาปัตยกรรมซอฟต์แวร์…….....
ระบบการลงทะเบียนมหาวิทยาลัย ก่อนการเปลี่ยนแปลง……..... 98
รูปที่ 5.8 กรณีทดสอบที่ 4 ภาพรวมสถาปัตยกรรมซอฟต์แวร์…….....
ระบบการลงทะเบียนมหาวิทยาลัย หลังการเปลี่ยนแปลง……..…….... 100

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

สถาปัตยกรรมซอฟต์แวร์ [1] เป็นกลุ่มของโครงสร้างที่จำเป็นต่อซอฟต์แวร์ ประกอบด้วยส่วนประกอบต่างๆ ที่มีความสัมพันธ์กัน โดยสถาปัตยกรรมซอฟต์แวร์มีความสำคัญในการควบคุมความซับซ้อนของซอฟต์แวร์ อีกทั้งยังสามารถนำไปใช้เพื่อปรับปรุงคุณภาพซอฟต์แวร์ก่อนที่จะเริ่มต้นกระบวนการพัฒนาได้ ทั้งนี้ การออกแบบสถาปัตยกรรมซอฟต์แวร์ที่ดีนั้น สามารถทำให้กระบวนการปรับปรุงซอฟต์แวร์ในช่วงของการบำรุงรักษาสามารถทำได้ง่ายขึ้นอีกด้วย

ขั้นตอนการออกแบบสถาปัตยกรรมซอฟต์แวร์โดยทั่วไป มักเริ่มจากการนำข้อมูลที่ระบุไว้ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ (Software Requirements Specification) มาวิเคราะห์ความต้องการเชิงหน้าที่ (Functional Requirements) เพื่อการออกแบบโครงร่างของสถาปัตยกรรมซอฟต์แวร์โดยรวม ทำให้ได้มาซึ่งแบบจำลองสถาปัตยกรรมซอฟต์แวร์ที่มีความเหมาะสมกับข้อกำหนดและข้อจำกัดต่างๆ ที่ได้ถูกระบุไว้ จากนั้นแบบจำลองสถาปัตยกรรมซอฟต์แวร์ที่ได้จะผ่านขั้นตอนในการประเมินความสามารถ ซึ่งถูกประเมินจากคุณลักษณะเชิงคุณภาพของซอฟต์แวร์ (Software Quality Attributes) [2] มีวิธีในการตรวจสอบความสามารถที่แตกต่างกัน ยกตัวอย่างเช่น การให้นักออกแบบหรือผู้ที่มีประสบการณ์เกี่ยวข้องกับโครงการซอฟต์แวร์ที่มีลักษณะของข้อกำหนดข้อจำกัด หรือการระบุความต้องการเชิงคุณภาพในลักษณะเดียวกันมาช่วยประเมินแบบจำลองสถาปัตยกรรมซอฟต์แวร์ที่ได้ออกแบบว่าองค์ประกอบที่ระบุไว้ มีคุณสมบัติเพียงพอที่จะตอบสนองตามข้อกำหนดข้างต้นหรือไม่ ในกรณีซอฟต์แวร์หรือระบบที่ทำการพัฒนาขึ้นมีความซับซ้อน การออกแบบสถาปัตยกรรมซอฟต์แวร์ย่อมมีผู้ที่เกี่ยวข้องจำนวนมาก ดังนั้น ในขั้นตอนของการออกแบบมักจะมีการตรวจสอบและปรับแก้ไขรายละเอียดเพิ่มเติม เพื่อที่จะนำมาซึ่งการพัฒนาซอฟต์แวร์ที่มีความถูกต้องเหมาะสม ตรงกับความต้องการของผู้ใช้งานมากที่สุด รายละเอียดดังกล่าวสามารถแสดงได้ดังรูปที่ 1.1



รูปที่ 1.1 ภาพรวมการออกแบบสถาปัตยกรรมซอฟต์แวร์ [3]

จากการศึกษางานวิจัยที่เกี่ยวข้องกับวิวัฒนาการของสถาปัตยกรรมซอฟต์แวร์ [4] มักจะพบปัญหาหรือข้อจำกัดต่างๆ ที่เกิดขึ้นก่อนหรือระหว่างการดำเนินการในการจัดทำโครงร่างแบบจำลองสถาปัตยกรรมซอฟต์แวร์ ดังต่อไปนี้

- ในการออกแบบสถาปัตยกรรมซอฟต์แวร์ที่มีส่วนประกอบจำนวนมาก หรือมีความซับซ้อนสูงนั้น เป็นการยากที่นักออกแบบจะสามารถแยกความแตกต่างระหว่างสถาปัตยกรรมซอฟต์แวร์แต่ละเวอร์ชันที่ออกแบบในรายละเอียดได้
- นักออกแบบไม่สามารถตรวจสอบข้อกำหนดความต้องการซอฟต์แวร์ที่เกี่ยวข้องคุณลักษณะเชิงคุณภาพ ตลอดจนส่วนประกอบหรือชุดคำสั่งที่อาจได้รับผลกระทบจากการเปลี่ยนแปลงส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ได้
- บ่อยครั้งมักเกิดข้อผิดพลาดในการตรวจสอบความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์โดยนักออกแบบเอง
- ขาดเครื่องมือที่ใช้ในการตรวจจับความเปลี่ยนแปลงหรือตรวจจับผลกระทบที่เกิดจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์

จากปัญหาที่ได้กล่าวมาข้างต้น ในงานวิจัยนี้จึงมุ่งเน้นในการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์เพื่อแก้ไขปัญหาที่พบระหว่างขั้นตอนการออกแบบสถาปัตยกรรมซอฟต์แวร์ โดยการจัดทำเครื่องมือเพื่อใช้สำหรับตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์สองเวอร์ชันที่เกี่ยวข้องกัน ผลกระทบจากการเปลี่ยนแปลงดังกล่าว หมายถึง รายการข้อกำหนดความต้องการซอฟต์แวร์ที่เกี่ยวข้อง คุณลักษณะเชิงคุณภาพ รวมถึงส่วนประกอบหรือชุดคำสั่งที่อาจได้รับผลกระทบจากการเปลี่ยนแปลงส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ ผลลัพธ์ที่ได้จากการนำเครื่องมือที่พัฒนาขึ้นไปใช้สามารถแสดงในรูปแบบรายงานสรุปส่วนประกอบที่ได้รับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ ซึ่งรวบรวมข้อมูลรายละเอียดของข้อกำหนดความต้องการซอฟต์แวร์ คุณลักษณะเชิงคุณภาพ หรือชุดคำสั่งใดๆ ที่เกี่ยวข้อง นอกจากนี้ เครื่องมือที่พัฒนาขึ้นยังสามารถแสดงผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ในรูปแบบรายงานสรุปผลความเปลี่ยนแปลง และรายงานแสดงรายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ได้ ทั้งนี้ ข้อมูลที่ได้จากการทำงานของเครื่องมือดังกล่าวสามารถนำไปใช้ในการวิเคราะห์ผลกระทบต่อการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์สำหรับผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์ ซึ่งเป็นประโยชน์ต่อการตัดสินใจในการพัฒนาสถาปัตยกรรมซอฟต์แวร์อย่างเหมาะสม ทำให้ได้สถาปัตกรรมซอฟต์แวร์ที่มีความสอดคล้องกับข้อจำกัดต่างๆ อีกทั้ง ยังมีความครบถ้วนตามข้อกำหนดความต้องการซอฟต์แวร์ที่ได้กำหนดไว้ นั่นเอง

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อออกแบบวิธีการตรวจจับความเปลี่ยนแปลงและผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
2. เพื่อพัฒนาเครื่องมือที่สนับสนุนวิธีการตรวจจับความเปลี่ยนแปลงและผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์

1.3 ขอบเขตของการวิจัย

1. พัฒนาเครื่องมือที่มีหน้าที่การทำงานหลักแบ่งเป็น 3 ส่วน รายละเอียดดังต่อไปนี้
    1) ส่วนจัดการข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ สามารถรับข้อมูลนำเข้าโดยผู้ใช้งานกรอกรายละเอียดข้อมูลคำอธิบายโครงการเบื้องต้น รายการข้อกำหนดความต้องการของโครงการ อีกทั้ง สามารถนำเข้าและ

ส่งออกข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการที่ได้บันทึกไว้ไปใช้งานในภายหลังได้

2) ส่วนจัดการคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ สามารถบันทึกข้อมูลคำอธิบายการออกแบบให้กับส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ได้

3) ส่วนการตรวจจับผลกระทบและสรุปผลความเปลี่ยนแปลงของสถาปัตยกรรมซอฟต์แวร์

สามารถตรวจจับผลกระทบและสรุปผลความเปลี่ยนแปลงของสถาปัตยกรรมซอฟต์แวร์สองเวอร์ชันที่เกี่ยวข้องกัน โดยผลกระทบ หมายถึง รายการข้อกำหนดความต้องการ คุณสมบัติเชิงคุณภาพ ส่วนประกอบหรือชุดคำสั่งที่ได้รับผลกระทบจากการเปลี่ยนแปลง

2. ข้อมูลนำเข้าที่ใช้ในส่วนการตรวจจับผลกระทบและสรุปผลความเปลี่ยนแปลงของสถาปัตยกรรมซอฟต์แวร์ ใช้แฟ้มข้อมูลที่จัดเก็บในลักษณะโครงสร้างภาษาเอ็กซ์เอดีแอลเวอร์ชัน 2.0 จำนวน 2 แฟ้ม

3. ส่วนประกอบพื้นฐานของสถาปัตยกรรมซอฟต์แวร์ที่ใช้ในการตรวจจับความเปลี่ยนแปลงมีทั้งหมด 4 ส่วน คือ

1) ส่วนประกอบ (Components)
2) ตัวเชื่อม (Connectors)
3) จุดเชื่อมต่อ (Interfaces)
4) เส้นเชื่อม (Links)

โดยมีลักษณะของการเปลี่ยนแปลงที่ใช้ตรวจจับ คือ การเพิ่ม (Add) การลบออก (Remove) และการแก้ไข (Modify) ซึ่งลักษณะของการเปลี่ยนแปลงที่ใช้ตรวจจับการแก้ไขข้อมูลส่วนประกอบ มีทั้งหมด 7 ส่วน คือ

1) คำอธิบาย (Description)
2) เส้นเชื่อม (Links)
3) จำนวนของจุดเชื่อมต่อ (Interface Size)
4) ประเภทส่วนประกอบ (Component Type)
5) ประเภทตัวเชื่อม (Connector Type)
6) ประเภทจุดเชื่อมต่อ (Interface Type)
7) คำอธิบายการออกแบบ (Design Description)

4. การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ แบบจำลองสถาปัตยกรรมซอฟต์แวร์ที่นำมาใช้ตรวจจับต้องได้รับการกำหนดคำอธิบายการออกแบบให้กับส่วนประกอบจากขั้นตอนการจัดการคำอธิบายการออกแบบ

5. การตรวจจับและสรุปผลความเปลี่ยนแปลง เป็นการตรวจจับความเปลี่ยนแปลงด้านโครงสร้างสถาปัตยกรรมซอฟต์แวร์เท่านั้น ทั้งนี้ ผลสรุปดังกล่าว ไม่ครอบคลุมในส่วนของการวิเคราะห์ด้านคุณภาพ

6. การตรวจสอบความถูกต้องของผลลัพธ์จากเครื่องมือที่พัฒนา ทำโดยการทดสอบจากการกำหนดกรณีทดสอบทั้งหมด จำนวน 4 กรณีทดสอบ ซึ่งแยกตามลักษณะของกรณีทดสอบได้ดังนี้

   1) ตัวอย่างการเปลี่ยนแปลงของสถาปัตยกรรมซอฟต์แวร์ที่เกี่ยวข้องกัน ซึ่งมีเงื่อนไขสำหรับการตรวจจับผลกระทบจากการเปลี่ยนแปลงที่ได้กำหนดไว้ครบทุกเงื่อนไข 1 กรณีทดสอบ
   2) ตัวอย่างการเปลี่ยนแปลงของสถาปัตยกรรมซอฟต์แวร์ที่เกี่ยวข้องกันแบบทั่วไป 3 กรณีทดสอบ

7. เครื่องมือสามารถแสดงรายงานสรุปผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ โดยมีรายละเอียดแสดงในรายงาน คือ รายการข้อกำหนดความต้องการซอฟต์แวร์ที่เกี่ยวข้อง คุณลักษณะเชิงคุณภาพ ส่วนประกอบหรือชุดคำสั่งใดๆ ที่ได้รับผลกระทบเป็นอย่างน้อย และยังแสดงรายงานสรุปผลความเปลี่ยนแปลง และรายงานแสดงรายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ได้

8. พัฒนาเครื่องมือเป็นแอปพลิเคชันซึ่งพัฒนาจากภาษาจาวา (Java) และทำงานบนระบบปฏิบัติการไมโครซอฟท์วินโดว์เอ็กซ์พี (Microsoft Windows XP) ขึ้นไป

1.4 ลำดับขั้นตอนในการนำเสนองานวิจัย

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 6 บท ดังต่อไปนี้ บทที่ 1 บทนำกล่าวถึงความเป็นมาและความสำคัญของปัญหา รวมถึงวัตถุประสงค์ของการวิจัย ขอบเขตของการวิจัย ประโยชน์ที่คาดว่าจะได้รับ บทที่ 2 กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้อง บทที่ 3 นำเสนอการวิเคราะห์และการออกแบบการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ บทที่ 4 การออกแบบและการพัฒนาเครื่องมือ บทที่ 5 กล่าวถึงการทดสอบเครื่องมือ บทที่ 6 เป็นบทสรุปผลการวิจัย

1.5 บทความวิชาการที่ได้รับการตีพิมพ์

ผลงานวิจัยที่ได้จัดทำร่วมกันโดยคณะผู้วิจัยเป็นบทความวิชาการระดับนานาชาติ คือ บทความวิชาการเรื่อง “EFFECT DETECTION OF SOFTWARE ARCHITECTURE CHANGES IN xADL” ซึ่งได้รับการคัดเลือกเพื่อนำเสนอและตีพิมพ์ในงาน “The 2013 IEANG International Conference on Software Engineering Hong Kong” ระหว่างวันที่ 13-15 มีนาคม 2556 ณ โรงแรม The Royal Garden Hotel เมืองเกาลูน เขตปกครองพิเศษฮ่องกง ประเทศสาธารณรัฐประชาชนจีน

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1. ทฤษฎีที่เกี่ยวข้อง

2.1.1. **สถาปัตยกรรมซอฟต์แวร์** (Software Architecture)

หมายถึง ภาพรวมของระบบซอฟต์แวร์ ซึ่งเป็นเหมือนกับพิมพ์เขียวของระบบ โดยนำไปใช้สำหรับการแจกแจงงาน ในกลุ่มทีมงานผู้ออกแบบและผู้พัฒนาระบบ นอกจากนี้ สถาปัตยกรรมซอฟต์แวร์ยังมีส่วนเกี่ยวข้องโดยตรงกับคุณลักษณะเชิงคุณภาพของซอฟต์แวร์ในหลายด้าน เช่น ด้านประสิทธิภาพ การปรับปรุง การบำรุงรักษา ความปลอดภัยของซอฟต์แวร์ เป็นต้น โดยโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่ได้กำหนดขึ้น ยังใช้สำหรับยืนยันความถูกต้องของแนวคิดและวิธีการออกแบบ สำหรับขั้นตอนในการวิเคราะห์ระบบซอฟต์แวร์ ได้อีกด้วย

ทั้งนี้ ระบบโดยส่วนใหญ่ถูกสร้างขึ้นเพื่อที่จะให้บริการในช่วงเวลายาวนาน อย่างไรก็ตาม การวางแผนเพื่อให้ระบบสามารถใช้งานในระยะเวลานานเป็นเรื่องยาก เนื่องจากการพัฒนาทางด้านเทคโนโลยีทั้งซอฟต์แวร์และฮาร์ดแวร์เป็นไปอย่างรวดเร็ว ด้วยเหตุนี้จึงเป็นสาเหตุทำให้การออกแบบสถาปัตยกรรมซอฟต์แวร์ในปัจจุบันมีความจำเป็นที่จะต้องสามารถรองรับการปรับเปลี่ยนในอนาคตได้ โดยการเปลี่ยนแปลงดังกล่าวจะต้องไม่ส่งผลกระทบต่อการทำงานโดยรวมของระบบ อย่างไรก็ตาม ในขณะที่การเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์สามารถปรับเปลี่ยนเพิ่มเติมขึ้นเมื่อเวลาผ่านไป ก็อาจจะกลายเป็นปัญหากับระบบได้ ในที่สุดจึงเป็นเรื่องยากที่จะทำให้เกิดการเปลี่ยนแปลง ยกตัวอย่างเช่น การเปลี่ยนแปลงซึ่งมีค่าใช้จ่ายมากเกินไป หรือใช้เวลาการดำเนินการนานทำให้ระบบหยุดการทำงาน หรือการเปลี่ยนแปลงที่อาจไม่ได้ตอบสนองเป้าหมายทางธุรกิจ คำถามสำหรับการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่พบได้บ่อย คือ วิธีการที่องค์กรสามารถนำมาจัดการความไม่แน่นอนในอนาคต เช่น การวางแผนสำหรับวิวัฒนาการของระบบในทางที่เพิ่มประสิทธิภาพการใช้งาน มีสิ่งที่จะต้องทำการพิจารณาจากข้อมูลพื้นฐานมาจากสองด้านหลัก ดังนี้

1) ด้านความคุ้มค่าเชิงธุรกิจ: จากการใช้วิธีการวางแผนเพื่อเพิ่มความสามารถของระบบในอนาคตโดยใช้การคาดการณ์อย่างเหมาะสม รวมถึงการเตรียมความพร้อม การจัดกลุ่มของความต้องการ การจัดลำดับความสำคัญของการเปลี่ยนแปลง เป็นต้น
2) ด้านการออกแบบ: จากการใช้วิธีการวิเคราะห์ความเสี่ยงในการปรับปรุงสถาปัตยกรรมซอฟต์แวร์ที่จะดำเนินการปรับปรุงเพิ่มเติมต้องคำนึงถึงทรัพยากรด้านต่างๆ ในปัจจุบันขององค์กร ยกตัวอย่างเช่น เทคโนโลยีที่นำมาใช้งาน ทรัพยากร

บุคคล หรือระยะเวลาในการดำเนินงาน รวมถึงวิธีการออกแบบระบบเพื่อให้เป็นที่น่าพอใจต่อผู้ที่เกี่ยวข้องหรือผู้ที่ได้รับผลกระทบต่อการเปลี่ยนแปลงโดยมีเป้าหมายเพื่อเพิ่มประสิทธิภาพและประสิทธิผลของการทำงานในปัจจุบัน เป็นต้น

2.1.2. **ภาษาเอดีแอล** (ADL - Architecture description language) [5]

ภาษาที่ใช้สำหรับระบุลักษณะโครงแบบของสถาปัตยกรรมซอฟต์แวร์ หรือระบบที่นักพัฒนาทำการออกแบบ โดยภาษาเอดีแอลนั้นยังสนับสนุนการกำหนดโครงสร้างของสถาปัตยกรรมซอฟต์แวร์ในลักษณะต่างๆ ซึ่งเป็นการกำหนดรูปแบบที่ชัดเจนโดยเป้าหมายของการออกแบบภาษานั้น ได้แก่ ทำให้ภาษาสามารถสนับสนุนการวิเคราะห์ อย่างน้อยในแง่ของความหมายเบื้องต้นของสถาปัตยกรรมซอฟต์แวร์ในลักษณะที่แตกต่างกันไป หรือการสนับสนุนกิจกรรมการออกแบบในโดเมนเฉพาะในอุตสาหกรรมใดอุตสาหกรรมหนึ่ง ยกตัวอย่างเช่น การนำภาษาเอดีแอลไปใช้ออกแบบสถาปัตยกรรมซอฟต์แวร์เกี่ยวกับการควบคุมการทำงานของถังบรรจุสารเคมีอันตราย เป็นต้น

2.1.3. **ภาษาเอ็กซ์เอดีแอล** (xADL - Highly-Extension architecture description language) [6]

เป็นหนึ่งในรูปแบบของภาษาที่ใช้สำหรับการระบุโครงสร้างของสถาปัตยกรรมซอฟต์แวร์หรือระบบ ถูกพัฒนาขึ้นโดย Institute for Software Research ที่มหาวิทยาลัย University of California เมือง Irvine มลรัฐแคลิฟอร์เนีย ประเทศสหรัฐอเมริกา ปัจจุบันภาษาเอ็กซ์เอดีแอล เป็นเวอร์ชัน 2.0 โดยมีความแตกต่างกับภาษาเอดีแอลอื่นๆ เนื่องจากมีการระบุโครงสร้างของสถาปัตยกรรมซอฟต์แวร์ในลักษณะกลุ่มโครงร่างของข้อมูลแบบเอ็กซ์เอ็มแอล (XML) [7] ทำให้เกิดความยืดหยุ่นในการปรับปรุงแก้ไข เพราะโครงสร้างสถาปัตยกรรมที่ถูกกำหนดโดยภาษาเอ็กซ์เอดีแอลนั้น สามารถที่จะรองรับการปรับปรุงแก้ไขได้จากเครื่องมือที่สนับสนุนการพัฒนาข้อมูลในรูปแบบเอ็กซ์เอ็มแอลได้นั่นเอง กลุ่มโครงร่างของภาษาเอ็กซ์เอดีแอลเวอร์ชัน 2.0 สนับสนุนการจัดสร้างโครงแบบในลักษณะดังต่อไปนี้

- สนับสนุนการพัฒนาส่วนประกอบของซอฟต์แวร์ในช่วงเวลาการออกแบบและการใช้งาน
- รองรับการพัฒนาสถาปัตกรรมซอฟต์แวร์ในประเภทต่างๆ
- การจัดการเปลี่ยนแปลงของข้อมูลในลักษณะการควบคุมเวอร์ชัน
- แยกประเภทของสถาปัตยกรรมซอฟต์แวร์ตามลักษณะของผลิตภัณฑ์

โดยปัจจุบันมีเครื่องมือที่สามารถรองรับและสนับสนุนการพัฒนาโครงร่างของสถาปัตยกรรมด้วยภาษาเอ็กซ์เอดีแอล ได้แก่ Apigen [8], xArch [9] และ ArchStudio 4 [10], ArchStudio 5 [11] และ Architecture-centric Development Environment [12] เป็นต้น

2.1.4. **เอ็กซ์เอ็มแอล, ดีทีดี และ เค้าร่างเอ็กซ์เอ็มแอล** (XML, DTDs and XML Schemas) [13]

เกิดขึ้นในปี ค.ศ. 1996 โดย “World Wide Web Consortium (W3C)” ได้มีการกำหนดมาตรฐานภาษาใหม่ ที่เรียกว่า “Extensible Markup Language” (XML) “เอ็กซ์เอ็มแอล” อ้างถึงตระกูลของเทคโนโลยีทั้งหมดที่เกี่ยวข้องกับข้อกำหนดตามเอ็กซ์เอ็มแอลเวอร์ชัน 1.0 โดยข้อกำหนดนี้มีการกำหนดหน้าที่ของภาษาสำหรับการระบุข้อมูลของเอกสารในรูปแบบข้อความโดยแบ่งแยกออกเป็นส่วนย่อยๆ ตามการกำหนดแถบของข้อมูล ที่เรียกว่า “tags” โดยภาษาเอ็กซ์เอ็มแอล เป็นภาษาที่มีการแบ่งแยกข้อมูลเป็นชั้น ตามการเปิดและปิดแถบของข้อมูล ซึ่งสามารถกำหนดคุณสมบัติต่างๆ เพิ่มเติมได้ นอกจากนี้ตามข้อกำหนดของภาษาเอ็กซ์เอ็มแอลเวอร์ชัน 1.0 ยังมีการกำหนดส่วนสำคัญของภาษาที่เรียกว่า “ดีทีดี” Document type definition (DTD) ทำให้ผู้พัฒนาโครงสร้างภาษาเอ็กซ์เอ็มแอล สามารถกำหนดวากยสัมพันธ์ (Syntax) ของเอกสาร ในคำจำกัดความที่เพิ่มมากขึ้น ยกตัวอย่างเช่น การกำหนดส่วนประกอบหรือคุณสมบัติใดที่ได้รับอนุญาตให้จัดเก็บในเอกสารดังกล่าวได้ เป็นต้น ส่วนต่อขยายเพิ่มเติมในภาษาเอ็กซ์เอ็มแอลนั้นเรียกว่า XML namespaces [14] ซึ่งอนุญาตให้ผู้พัฒนาโครงสร้างภาษาเอ็กซ์เอ็มแอล สามารถนำเข้าข้อมูลส่วนประกอบและคุณสมบัติต่างๆ จากการกำหนดส่วนประกอบที่หลากหลายได้

หลังจากที่มีการใช้งานภาษาเอ็กซ์เอ็มแอลกันอย่างแพร่หลาย W3C จึงได้มีการกำหนดเค้าร่างเอ็กซ์เอ็มแอล (XML Schema) ซึ่งเป็นการแทนที่ความสามารถของดีทีดีเดิมได้อย่างชัดเจน โดยเค้าร่างเอ็กซ์เอ็มแอลมีความยืดหยุ่นต่อการใช้งานตามแต่ผู้พัฒนาโครงสร้างภาษานำไปใช้กับระบบใด เค้าร่างเอ็กซ์เอ็มแอลยังมีความสามารถในการรองรับการรวมกันของประเภท (Compound types) ที่กำหนดขึ้น และการสืบทอดคุณสมบัติจากประเภทของข้อมูลพื้นฐาน (Type inheritance) ลักษณะคล้ายกันกับแนวคิดการออกแบบโปรแกรมเชิงวัตถุ ซึ่งทำให้ผู้พัฒนาสามารถใช้งานเค้าร่างเอ็กซ์เอ็มแอลที่มีคุณสมบัติเฉพาะบางอย่างแตกต่างไปจากประเภทข้อมูลพื้นฐาน นั่นเอง

2.1.5. **คุณลักษณะเชิงคุณภาพของซอฟต์แวร์** (Software Quality Attributes)

เป็นส่วนหนึ่งในความต้องการเชิงคุณภาพของซอฟต์แวร์ ซึ่งเป็นส่วนส่งเสริมต่อการทำงานเชิงหน้าที่ของซอฟต์แวร์ให้สามารถทำงานสำเร็จได้อย่างสมบูรณ์ ในการกำหนดคุณลักษณะเชิง

คุณภาพของซอฟต์แวร์ ตามมาตรฐาน ISO/IEC 9126: 2001 [15] ซึ่งเป็นมาตรฐานที่ได้รับการยอมรับในกลุ่มนักวิศวกรรมซอฟต์แวร์ แบ่งคุณลักษณะเชิงคุณภาพของซอฟต์แวร์ทั้งหมด ออกได้เป็น 6 ประเภท ดังนี้

1) หน้าที่การทำงาน (Functionality) คือ ลักษณะความสามารถที่ได้รับการกำหนดเป็นหน้าที่การทำงาน โดยจะต้องสอดคล้องกับความต้องการของผู้ใช้งาน ซึ่งตรวจสอบได้จากการนำผลิตภัณฑ์ซอฟต์แวร์ไปใช้
2) ความน่าเชื่อถือ (Reliability) คือ ลักษณะความสามารถของผลิตภัณฑ์ซอฟต์แวร์ ที่สามารถรักษาระดับของการปฏิบัติงานตามที่ระบุไว้ในเงื่อนไขหรือกรอบระยะเวลาของการใช้งาน
3) การใช้งาน (Usability) คือ ความสามารถของผลิตภัณฑ์ซอฟต์แวร์ที่จะทำให้ผู้ใช้งานได้รับการเรียนรู้จากการใช้งานภายใต้เงื่อนไขที่กำหนดไว้
4) ประสิทธิภาพ (Efficiency) คือ ความสามารถของผลิตภัณฑ์ซอฟต์แวร์ที่จะรักษาความสัมพันธ์ระหว่างระดับของการทำงานของซอฟต์แวร์และปริมาณของทรัพยากรที่ใช้ตามเงื่อนไขที่กำหนดไว้
5) การบำรุงรักษา (Maintainability) คือ คุณลักษณะของผลิตภัณฑ์ซอฟต์แวร์ ที่สามารถรองรับการปรับเปลี่ยน ตามความต้องการเชิงหน้าที่ หรือการปรับแก้ไขสภาพแวดล้อมที่เกี่ยวข้องเพิ่มเติม
6) ความสะดวกในการพกพา (Portability) คือ คุณลักษณะที่ผลิตภัณฑ์ซอฟต์แวร์มีความสามารถที่จะโอนย้ายมาจากสภาพแวดล้อมหนึ่งไปยังที่อื่นๆ

2.1.6. **ระบบซีวีเอส** (CVS - Concurrent Versions System) [16] **หรือ** Concurrent Versioning System

ถูกพัฒนาขึ้นครั้งแรกในปี ค.ศ. 1986 โดย Dick Grune นักพัฒนาซอฟต์แวร์ชาวเนเธอร์แลนด์ ในรูปแบบฟรีแวร์ที่มีลักษณะสถาปัตยกรรมแบบไคลเอ็น-เซิร์ฟเวอร์ ใช้ประโยชน์สำหรับการควบคุม ตามรอยการเปลี่ยนแปลงของคอนเทนต์หรือเอกสาร นิยมใช้กันอย่างแพร่หลายในกระบวนการพัฒนาซอฟต์แวร์ โดยคุณสมบัติหลัก คือ สามารถจัดเก็บ ติดตามรอยการเปลี่ยนแปลง ตรวจสอบความแตกต่างระหว่างเวอร์ชัน บันทึกข้อมูลการเปลี่ยนแปลงต่างๆ ที่เกี่ยวข้องกับคอนเทนต์หรือเอกสารได้ อีกทั้งยังสามารถช่วยให้ทีมผู้พัฒนาซอฟต์แวร์ที่มีลักษณะการใช้งานคอนเทนต์หรือเอกสารร่วมกัน ทำงานได้อย่างมีประสิทธิภาพ อีกด้วย ปัจจุบันนี้ ในส่วนของซอฟต์แวร์ทั่วไปในรูปแบบฟรีแวร์ โปรแกรมซีวีเอสยังถือว่าเป็นซอฟต์แวร์ที่ได้รับความนิยมเป็นอย่างมาก

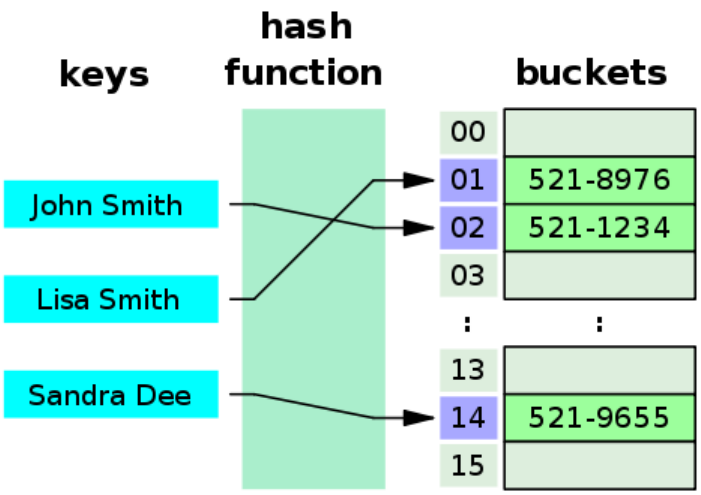
2.1.7. **ตารางแฮช** (Hash Table) [17]

คือ หนึ่งในโครงสร้างข้อมูล (Data Structure) มักใช้สำหรับการพัฒนาซอฟต์แวร์ โดยนอกเหนือจากนี้ยังมีลักษณะโครงสร้างข้อมูลแบบอื่นๆ ยกตัวอย่าง เช่น สแตก คิว ลิงค์ลิสต์ เป็นต้น คำว่า ตารางแฮช หรือ Hash Table เป็นหนึ่งในคำศัพท์ด้านการเขียนโปรแกรม นักเขียนโปรแกรมสามารถนำตารางแฮชมาใช้เก็บข้อมูลขนาดใหญ่ได้อย่างมีประสิทธิภาพ

1. กายวิภาคของตารางแฮช

ตารางแฮช คือ โครงสร้างข้อมูลที่ใช้เพื่อเก็บออบเจ็กต์ที่มีความสัมพันธ์ระหว่างค่าคีย์หรือตำแหน่งที่อยู่ของออบเจ็กต์ (Key) และค่าที่จัดเก็บ (Value) ตารางแฮชเป็นอาร์เรย์ที่เก็บพอยน์เตอร์โดยชี้ไปยังข้อมูล (Array of pointer เรียกย่อว่า AP) ข้อมูลจะอยู่ในรูปแบบของโครงสร้างข้อมูลที่ผู้ใช้นิยามขึ้นเอง (User Define Structure เรียกย่อว่า UDS) อันประกอบด้วยส่วนต่างๆ สามส่วน คือ ค่าคีย์, ค่าที่จัดเก็บและพอยน์เตอร์ที่ทำหน้าที่ชี้ไปยัง UDS โดย ค่าคีย์ ทำหน้าที่แยกแยะความแตกต่างให้แก่ข้อมูลใน UDS แต่ละตัวต้องมีค่าคีย์ที่ไม่ซ้ำกัน ค่าที่จัดเก็บ คือข้อมูลที่คู่กันกับค่าคีย์ ซึ่งภายในตารางแฮชหนึ่งๆ อาจมีข้อมูลซ้ำกันได้หรือจะเรียกได้ว่า ค่าคีย์ เป็นหมายเลขประจำตัวประชาชน ซึ่งอยู่คู่กันกับชื่อ คนแต่ละคนย่อมได้รับหมายเลขประจำตัวไม่ซ้ำกัน แต่บางคนชื่ออาจเหมือนกันได้ เป็นต้น

ตัวอย่างการจัดเก็บโครงสร้างข้อมูลแบบตารางแฮช สามารถแสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 ตัวอย่างการจัดเก็บโครงสร้างข้อมูลแบบตารางแฮช

2.2. งานวิจัยที่เกี่ยวข้อง

2.2.1 Composing Structural Views in xADL [18]

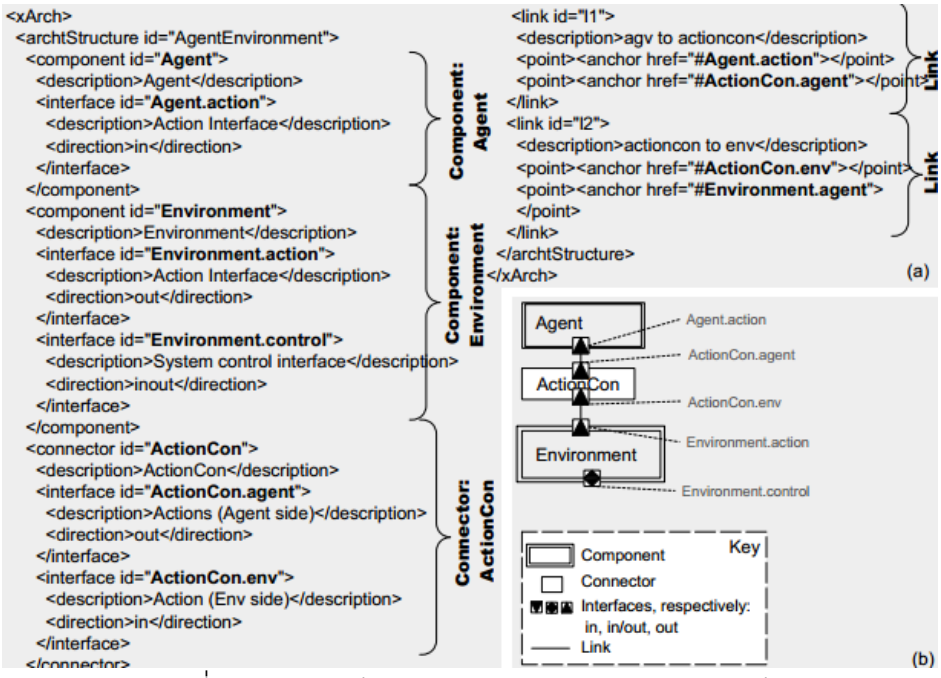
นำเสนอโดย Nelis Bouck´e, Alessandro Garcia และ Tom Holvoet ในปี ค.ศ. 2007 งานวิจัยนี้ได้อธิบายถึงโครงสร้างของภาษาเอ็กซ์เอดีแอล ซึ่งประกอบด้วยส่วนประกอบพื้นฐานทั้งหมด 4 ส่วน ดังนี้

1. ส่วนประกอบ (Components) หมายถึง ส่วนประกอบย่อยที่เชื่อมต่อกัน ประกอบเป็นโครงสร้างสถาปัตยกรรมซอฟต์แวร์ โดยส่วนประกอบย่อยดังกล่าว ยังมีรหัสซึ่งใช้เป็นสิ่งระบุตัวตน (Unique identifier) และ คำอธิบาย (Textual description) อีกทั้งยังอาจมีกลุ่มของจุดเชื่อมต่อ (Interfaces) รวมอยู่ด้วย

2. ตัวเชื่อม (Connectors) หมายถึง ตัวเชื่อมต่อระหว่างส่วนประกอบย่อยในการออกแบบสถาปัตยกรรมซอฟต์แวร์ เช่นเดียวกันกับส่วนประกอบ ตัวเชื่อม จะมีรหัสซึ่งใช้เป็นสิ่งระบุตัวตน คำอธิบาย และอาจมีกลุ่มของจุดเชื่อมต่อรวมอยู่ด้วยเช่นกัน

3. จุดเชื่อมต่อ (Interfaces) หมายถึง จุดที่ส่วนประกอบหรือตัวเชื่อมจะเชื่อมต่อไปยังส่วนประกอบอื่นๆ นอกจากนี้ จุดเชื่อมต่อ สำหรับภาษาเอ็กซ์เอดีแอล จะเรียกว่า พอร์ต (Port) สำหรับใช้กับส่วนประกอบ โดยมีรหัสซึ่งใช้เป็นสิ่งระบุตัวตน คำอธิบายและทิศทางของการเชื่อมต่อ โดยมีทิศทางที่สามารถระบุได้ คือ ทิศทางการเชื่อมต่อเข้า ออก หรือทิศทางเข้าและออก

4. เส้นเชื่อม (Links) หมายถึง เส้นทางที่ใช้ในการเชื่อมต่อ ระหว่างจุดเชื่อมต่อด้วยกัน ซึ่งส่วนประกอบของภาษาเอ็กซ์เอดีแอล ข้างต้น สามารถแสดงรายละเอียดได้ดังรูปที่ 2.2



รูปที่ 2.2 รายละเอียดส่วนประกอบของภาษาเอ็กซ์เอดีแอล [18]

2.2.2 A Highly-Extensible, XML-Based Architecture Description Language [6]

งานวิจัยนี้ เสนอโดย Eric M. Dashofy, Andre van der Hoek และ Richard N. Taylor ในปี ค.ศ. 2001 เป็นงานวิจัยที่อธิบายถึงความหมาย ความเป็นมา คำนิยามของส่วนประกอบต่างๆ ของภาษาเอ็กซ์เอดีแอล รวมทั้งมีส่วนของการยกตัวอย่างเครื่องมือที่ใช้ในการออกแบบสถาปัตยกรรมซอฟต์แวร์ผ่านส่วนต่อประสานกับผู้ใช้งาน เพื่อให้สามารถดำเนินการออกแบบหรือปรับปรุงแก้ไขโครงสร้างสถาปัตยกรรมซอฟต์แวร์ได้อย่างสะดวก ถูกต้องแม่นยำมากยิ่งขึ้น ซึ่งผลลัพธ์ที่ได้ คือ แบบจำลองโครงสร้างสถาปัตยกรรมซอฟต์แวร์ ที่มีข้อมูลการอธิบายลักษณะของส่วนประกอบต่างๆ รวมทั้งความสัมพันธ์กันระหว่างส่วนประกอบเหล่านั้นรวมอยู่ด้วย ทั้งนี้ การศึกษางานวิจัยดังกล่าว ยังสามารถนำมาใช้เป็นแนวทางเพื่อทำความเข้าใจกับการให้คำจำกัดความของส่วนประกอบ ความสามารถต่างๆ รวมถึงข้อจำกัดสำหรับภาษาที่ใช้ในการอธิบายส่วนประกอบโครงสร้างสถาปัตยกรรมซอฟต์แวร์โดยรวม ทำให้เกิดความเข้าใจเพิ่มเติมมากยิ่งขึ้น

งานวิจัยยังกล่าวถึง สถาปัตยกรรมซอฟต์แวร์ว่า เป็นการกำหนดโครงสร้างของระบบในมุมมองภาพกว้าง อีกทั้ง ยังเป็นการกำหนดเฉพาะส่วนประกอบที่สำคัญ โดยการระบุสถาปัตยกรรมซอฟต์แวร์ที่ดีนั้น ทำให้ผู้ที่เกี่ยวข้องทราบถึงส่วนประกอบและคุณสมบัติเบื้องต้นของระบบในมุมมองแบบเค้าร่างได้อีกด้วย ความหมายของคุณสมบัติซึ่งรวมไว้ในสถาปัตยกรรมซอฟต์แวร์ อาจหมายถึงลักษณะการเชื่อมต่อหรือส่วนประกอบต่างๆ ส่วนจัดเก็บข้อมูล เป็นต้น อย่างไรก็ตาม การออกแบบสถาปัตยกรรมซอฟต์แวร์นั้นมีส่วนสำคัญอย่างยิ่งสำหรับความสำเร็จต่อระบบที่ใช้ซอฟต์แวร์เป็นพื้นฐาน การมีตัวเลือกสำหรับสถาปัตยกรรมในลักษณะต่างๆนั้น ช่วยให้ผลิตภัณฑ์ที่ส่งมอบตรงกับความต้องการและง่ายต่อการแก้ไขปรับปรุงสำหรับความต้องการใหม่ แต่ในบางครั้ง สำหรับนักวิศกรรมซอฟต์แวร์ที่ขาดการฝึกฝนในการออกแบบโครงสร้างสถาปัตยกรรมอย่างเป็นทางการ ทำให้การนำรายละเอียดต่างๆไปใช้ในขั้นตอนการพัฒนาซอฟต์แวร์ขาดความชัดเจนได้ ยกตัวอย่างเช่น ขาดการวิเคราะห์ข้อจำกัดของซอฟต์แวร์ในขั้นตอนการออกแบบหรือการออกแบบไม่สมบูรณ์ แม้กระทั่งการออกแบบที่รายละเอียดต่างๆไม่สอดคล้องกัน เป็นต้น ซึ่งปัญหาสำคัญที่พบอีกข้อหนึ่ง คือ การขาดเครื่องมือสำหรับการออกแบบสถาปัตยกรรมซอฟต์แวร์ที่เหมาะสมกับนักออกแบบสถาปัตยกรรมซอฟต์แวร์

จากปัญหาที่พบ จึงมีนักวิจัยส่วนหนึ่งในวงการอุตสาหกรรมและวงการการศึกษานำเสนอการนิยามเครื่องหมายที่เป็นทางการ ในรูปแบบของการวิเคราะห์ในกระบวนการออกแบบสถาปัตยกรรมซอฟต์แวร์ขึ้น โดยทั่วไปก็หมายความรวมไปถึง ภาษาเอดีแอล ซึ่งเป็นเครื่องหมายหรือภาษาที่ใช้สำหรับอธิบายลักษณะของซอฟต์แวร์ อีกทั้งยังมีเครื่องมือที่สนับสนุนการวิเคราะห์ภาษา การแปลงภาษาให้อยู่ในรูปแผนภาพ ภาษาเอดีแอล ยกตัวอย่าง เช่น Aesop, Adage, Meta-H, C2, Rapide, SADL, UniCon, และ Wright เป็นต้น โดยทุกภาษาล้วนแล้วแต่มี

วัตถุประสงค์เพื่อใช้ในการออกแบบสถาปัตยกรรมซอฟต์แวร์ทั้งสิ้น ซึ่งสามารถแยกตามลักษณะความสามารถได้ ดังนี้ ภาษา Aesop รองรับการออกแบบซอฟต์แวร์อิเล็คทรอนิกส์สำหรับการสื่อสาร การบังคับ นำร่อง อากาศยาน เช่น เครื่องบิน ยานอวกาศ หรือดาวเทียม, ภาษา Meta-H รองรับการออกแบบซอฟต์แวร์สำหรับการควบคุมอากาศยาน, ภาษา C2 สนับสนุนการออกแบบซอฟต์แวร์ที่มีส่วนต่อประสานในลักษณะรูปแบบของการส่งข้อความระหว่างกัน, ภาษา Rapide ใช้สำหรับออกแบบซอฟต์แวร์โดยการจำลองผลลัพธ์, ภาษา SADL สนับสนุนการนิยามสถาปัตกรรมซอฟต์แวร์เบื้องต้น, ภาษา UniCon ใช้สำหรับเป็นตัวแปลภาษาในระดับบนของการออกแบบสถาปัตยกรรมซอฟต์แวร์ ประกอบไปด้วยส่วนประกอบและประเภทของการเชื่อมต่อ, ภาษา Wright สนับสนุนการวิเคราะห์สำหรับการต่อประสานระหว่างส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ เป็นต้น

การที่นักวิจัยหรือผู้ที่นำภาษาเอดีแอลไปใช้งานอย่างหลากหลายในลักษณะต่างๆ ดังที่กล่าวมาข้างต้น ย่อมมีข้อดี คือ สามารถนำไปประยุกต์ใช้งานในแต่ละมุมมองโดยการกำหนดความสามารถของภาษาให้สามารถนำไปใช้แก้ไขปัญหาที่พบได้ ซึ่งจุดกำเนิดของภาษาก็มาจากการพัฒนาซอฟต์แวร์เช่นกัน ซึ่งหมายถึงมีการกำหนดปัญหา วิเคราะห์แนวทางในการแก้ไขปัญหา จากนั้นจึงออกแบบส่วนที่ใช้สำหรับแก้ไขปัญหา แต่ ในอีกมุมหนึ่งการนำภาษาเอดีแอล ไปพัฒนาเพิ่มเติมตามแต่ละมุมมอง เพื่อแก้ไขตามแต่ละปัญหาที่มีความหลากหลายและจำเพาะเจาะจงในลักษณะดังกล่าวนั้น หนึ่งในสิ่งที่เป็นข้อเสียที่สำคัญ คือ ไม่มีการนำกลับมาใช้ใหม่ของส่วนการทำงานที่เป็นพื้นฐานของเครื่องมือที่สามารถนำมาใช้ร่วมกันได้ เช่น ส่วนต่อประสานของการแสดงผลในแต่ละภาษาเอดีแอล ทำให้สูญเสียทรัพยากรในการศึกษาและพัฒนาเพิ่มเติมมากขึ้น โดยภาษาเอดีแอล แต่ละแบบไม่สามารถแลกเปลี่ยนหรือแบ่งปันส่วนการทำงานใดๆ ร่วมกันได้ จึงมีแนวคิดในการปรับปรุงภาษาเอดีแอลที่มีความสามารถเพิ่มเติมในการแลกเปลี่ยนข้อมูลในลักษณะรูปแบบภาษากลางขึ้น ยกตัวอย่างเช่น ภาษาที่สามารถใช้งานระหว่างเครื่องมือที่แตกต่างกันได้ ซึ่งภาษาที่สามารถนำมาปรับใช้กับแนวคิดนี้ คือ ภาษาเอ็กซ์เอดีแอล

ภาษาเอ็กซ์เอดีแอล เป็นภาษาที่ใช้ในการอธิบายลักษณะของสถาปัตยกรรมซอฟต์แวร์แบบหนึ่ง ถูกพัฒนาขึ้นโดยกลุ่มนักวิจัย ผ่านการสนับสนุนงบประมาณจาก Defense Advanced Research Project Agency, Rome Laboratory, Air Force Material Command และ USAF ซึ่งหน่วยงานทั้งหมดอยู่ภายใต้การควบคุมโดยรัฐบาลของประเทศสหรัฐอเมริกา โดยกลุ่มนักวิจัยได้นำความสามารถและข้อกำหนดพื้นฐานต่างๆ มาจากภาษาเอดีแอล ในงานวิจัยนี้ได้อธิบายถึงความสามารถหลักของภาษาเอ็กซ์เอดีแอล ที่มา รวมถึงการพัฒนาทางด้านเทคนิค โดยเป้าหมายของการออกแบบภาษานั้น ได้แก่ การทำให้ภาษาสามารถสนับสนุนการวิเคราะห์ อย่างน้อยในแง่ของความหมายเบื้องต้นของสถาปัตยกรรมซอฟต์แวร์ในลักษณะที่แตกต่างกันไป หรือการ

สนับสนุนกิจกรรมการออกแบบในโดเมนเฉพาะในอุตสาหกรรมใดอุตสาหกรรมหนึ่ง ยกตัวอย่าง เช่น นำไปใช้ออกแบบสถาปัตยกรรมซอฟต์แวร์เกี่ยวกับการควบคุมการทำงานของถังบรรจุสารเคมีอันตราย โดยเป้าหมายหลักของการออกแบบภาษาที่สำคัญที่สุด ก็คือ เป็นภาษาที่สามารถแลกเปลี่ยนรูปแบบระหว่างเครื่องมือที่ใช้ในการพัฒนาสถาปัตยกรรมซอฟต์แวร์อย่างหลากหลายได้ นั่นเอง ความสามารถหลักของภาษาเอ็กซ์เอดีแอล ที่งานวิจัยได้กล่าวถึง อาทิเช่น

1. ด้วยโครงสร้างภาษาที่ถ่ายทอดมาจากพื้นฐานของเค้าร่างเอ็กซ์เอ็มแอล จึงสามารถเพิ่มเติมส่วนต่อขยายได้ ทำให้มีความยืดหยุ่น (Flexible annotation mechanism)
2. มีเครื่องมือสนับสนุนในการเรียงลำดับ (Sorting) การจัดการ (Manipulating) และการแบ่งปัน (Sharing) ระหว่างภาษาเอ็กซ์เอดีแอล ด้วยกันเอง
3. มีการแยกความแตกต่างของการอธิบายสถาปัตยกรรมในช่วงการออกแบบ (Architecture prescription) และการอธิบายสถาปัตยกรรมในช่วงเวลานำไปใช้งาน (Architecture description) เพิ่มเติม
4. มีการจัดเตรียมส่วนต่อขยายเพิ่มเติมที่เป็นประโยชน์ต่อการอธิบายสถาปัตยกรรมซอฟต์แวร์ ยกตัวอย่างเช่น การจัดการองค์ประกอบ (Configuration Management) และการจับคู่ส่วนประกอบต่างๆ กับการ Implementations ชุดคำสั่งของภาษาจาวา

โดยภาษาเอ็กซ์เอดีแอล ที่ได้กล่าวถึงนั้น มีพื้นฐานการเก็บข้อมูลในรูปแบบโครงสร้างของข้อมูลประเภทเอ็กซ์เอ็มแอล เหมือนกับภาษาเอดีแอลอื่นๆ ยกตัวอย่างเช่น ภาษา ADML [19] เป็นต้น ซึ่งใช้วิธีการกำหนดเอ็กซ์เอ็มแอลดีทีดี ทำให้เครื่องมือต่างๆ สามารถนำโครงสร้างของข้อมูลประเภทเอ็กซ์เอ็มแอลนี้ไปใช้งานร่วมกันได้ ได้แก่ เครื่องมือ COTS สามารถรองรับการสร้าง การแก้ไข การเปลี่ยนแปลง การตรวจสอบ การจัดเก็บ และการปรับเปลี่ยนเอกสารประเภทเอ็กซ์เอ็มแอลนี้ ได้อย่างสะดวก นอกจากนี้ ยังมี APIs อื่นๆ ที่สนับสนุนข้อมูลประเภทเอ็กซ์เอ็มแอล อีกมากมาย ยกตัวอย่างเช่น DOM [20], SAX [21] และ JDOM [22] เป็นต้น

ภาษาเอ็กซ์เอดีแอล เวอร์ชันในปัจจุบันคือ เวอร์ชัน 2.0 ถูกพัฒนาขึ้นที่ University of California เมือง Irvine มลรัฐแคลิฟอร์เนีย ประเทศสหรัฐอเมริกา ให้อยู่ในรูปแบบภาษาเอดีแอลที่สามารถแลกเปลี่ยนกันระหว่างเครื่องมือที่ใช้พัฒนาภาษาได้ โดยมีข้อดี คือ สามารถรองรับการออกแบบสถาปัตยกรรมซอฟต์แวร์ที่หลากหลาย และสามารถจับคู่ส่วนประกอบต่างๆ กับการ Implementations ชุดคำสั่งของภาษาจาวาได้ โดยการจัดโครงสร้างของภาษาเอ็กซ์เอดีแอล 2.0 สามารถแสดงตามจุดประสงค์ได้ดังตารางที่ 2.1

ตารางที่ 2.1 แสดงโครงสร้างของภาษาเอ็กซ์เอดีแอล 2.0 แสดงตามจุดประสงค์การออกแบบ [6]

Purpose	Schema	Features Provided
Architecture Modeling - Description and Prescription	xArch Instance	Component, connector, interface, and link instances; arbitrary groups; hierarchical construction.
	Structure & Types	Design-time architectural prescription; architectural structure (components, connectors, interfaces, and links), programming-language style types-and-instances model; hierarchical construction via types.
Instantiatable Architectures	Implementation	Abstract placeholder for implementation information for components and connectors.
	Java Implementation	Java-specific implementation information for components and connectors.
Architecture Configuration Management / Product Family Architectures	Options	Optional components, connectors, and links.
	Variants	Variant component and connector types.
	Versions	Version graphs for components, connectors, and interfaces.

จากวัตถุประสงค์ในการจัดโครงสร้างของภาษาเอ็กซ์เอดีแอล 2.0 ที่แสดงข้างต้นนั้น สามารถแบ่งออกเป็น 3 ส่วน ได้แก่

1) การอธิบายสถาปัตยกรรมในช่วงการออกแบบและช่วงเวลานำไปใช้งาน (Architecture Modeling Description and Prescription)

ภาษาเอ็กซ์เอดีแอล 2.0 ถูกกำหนดโดยมีส่วนประกอบรวมกันจาก xArch instances schema และเอ็กซ์เอดีแอล 2.0 structure & types schema โดยทำการปรับปรุงจากเวอร์ชันในการกำหนดการแยกความแตกต่างของการอธิบายสถาปัตยกรรมในช่วงการออกแบบ (Architecture prescription) และการอธิบายสถาปัตยกรรมในช่วงเวลานำไปใช้งาน (Architecture description) เพิ่มเติมขึ้นมา โดยในช่วงการออกแบบสถาปัตยกรรม หมายถึง การออกแบบส่วนประกอบสำหรับสถาปัตยกรรมซอฟต์แวร์ใดๆ แต่ช่วงเวลานำไปใช้งาน หมายถึง ส่วนประกอบที่ได้ออกแบบมานั้น ถูกสร้างขึ้นเพื่อนำไปใช้งานจริง เป็นต้น

2) การจับคู่ส่วนประกอบต่างๆ กับการ Implementations ชุดคำสั่งของภาษาจาวา (Implementation Mappings)

เป็นส่วนของการจับคู่ส่วนประกอบต่างๆ กับการ Implementations ชุดคำสั่งในภาษาจาวา ซึ่งจะสามารถทำให้นักออกแบบสามารถนำแบบจำลองสถาปัตยกรรมซอฟต์แวร์ที่ได้ออกแบบไว้ไปใช้งานได้โดยตรง ซึ่งภาษาเอ็กซ์เอดีแอล 2.0 ก็สามารถที่จะทำการจับคู่ส่วนประกอบต่างๆ กับการ Implementations ชุดคำสั่งของการพัฒนาโปรแกรมได้ในหลากหลายภาษา

3) ส่วนการจัดการองค์ประกอบสถาปัตยกรรม (Architecture Configuration Management)

ปัญหาสำคัญของการออกแบบสถาปัตยกรรมซอฟต์แวร์ที่ยืดหยุ่น โดยมีความถูกต้อง คือ สถาปัตยกรรมซอฟต์แวร์ในเวอร์ชันต่างๆ ที่ออกแบบมานั้น จะต้องสามารถจัดการข้อมูลพื้นฐานได้ ซึ่งการจัดการข้อมูลที่ได้กล่าวมานี้ หมายถึง การบริหารจัดการวิวัฒนาการของสถาปัตยกรรมซอฟต์แวร์ (Architecture-based configuration management) [23] ที่เป็นแนวความคิดของการจัดการที่จะสามารถนำมาแก้ไขปัญหาข้างต้นได้ นั่นเอง ซึ่งใช้แนวความคิดพื้นฐานมาจากการจัดการความเปลี่ยนแปลง คือ การจัดเก็บข้อมูลการเปลี่ยนแปลงที่สำคัญต่างๆ ยกตัวอย่างเช่น หมายเลขเวอร์ชัน และความแตกต่างที่เกิดขึ้นระหว่างเวอร์ชัน เป็นต้น

2.2.3 A rationale-based architecture model for design traceability and reasoning [24]

งานวิจัยนี้ นำเสนอโดย Antony Tang และ Yan Jin, Jun Han ในปี ค.ศ. 2007 โดยภาพรวมงานวิจัยได้กล่าวถึง ระบบขนาดใหญ่ที่มักจะมีช่วงเวลาในการใช้งานที่ยาวนาน ทั้งประกอบด้วยองค์ประกอบที่เกี่ยวข้องหลายอย่าง การตรวจสอบและการบำรุงรักษาระบบเหล่านี้ ผู้ที่ดำเนินการนั้นจะต้องมีความเข้าใจในการออกแบบสถาปัตยกรรมซอฟต์แวร์เป็นอย่างดี ผู้จัดทำงานวิจัยได้กล่าวถึงความสำคัญของหลักการออกแบบสถาปัตยกรรมซอฟต์แวร์โดยใช้เหตุผลในออกแบบเป็นพื้นฐาน ว่าจะทำให้ระบบที่ออกแบบขึ้น สามารถทำความเข้าใจที่มาและความสำคัญของส่วนประกอบ รวมถึงความสัมพันธ์กันของส่วนประกอบต่างๆ ได้ง่ายมากยิ่งขึ้น การขาดหลักของการออกแบบโดยใช้เหตุผลในออกแบบเป็นพื้นฐานดังกล่าว จะทำให้เกิดความยากลำบากที่จะตรวจสอบความไม่สอดคล้องกันหรือความขัดแย้งกันของสถาปัตยกรรมซอฟต์แวร์ที่ออกแบบขึ้น ปัญหาดังกล่าวสามารถแก้ไขได้ โดยผู้ดำเนินงานวิจัยได้นำเสนอแบบจำลองในการออกแบบสถาปัตยกรรมซอฟต์แวร์ซึ่งใช้หลักของการระบุเหตุผลในการออกแบบเป็นพื้นฐาน เพื่ออ้างอิงไปยัง

ส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ที่มีความสัมพันธ์ระหว่างกันและนำมาเชื่อมโยงกับเหตุผลในการออกแบบส่วนประกอบเหล่านั้น เพื่อแก้ไขปัญหาที่เกิดขึ้น

ทั้งนี้ การนำเสนอแบบจำลองในการแก้ไขปัญหาดังกล่าว ยังสามารถสนับสนุนการตามรอยวิเคราะห์ผลกระทบต่อการเปลี่ยนแปลง การวิเคราะห์ต้นเหตุของผลกระทบที่เกิดขึ้น และเครื่องมือที่นำมาใช้งานยังมีความสามารถในการจัดเก็บ การจำลองลักษณะของการตามรอยแบบจำลองที่นำเสนอได้ นอกจากนี้ ผู้ดำเนินงานวิจัย ได้ยกตัวอย่างการนำแบบจำลองไปทดลองใช้งานในอุตสาหกรรมซอฟต์แวร์ โดยใช้ระบบการจ่ายเงินแบบอิเล็กทรอนิกส์เป็นกรณีตัวอย่าง ซึ่งจะสามารถทำให้นักสถาปนิกซอฟต์แวร์สามารถเข้าใจถึงความสำคัญของการนำเหตุผลในการออกแบบสถาปัตยกรรมไปประยุกต์ใช้งานกับการทำงานจริงได้ดีมากขึ้น

ปัญหาที่มักเกิดขึ้น เมื่อขาดการนำหลักของการระบุเหตุผลในการออกแบบไปประยุกต์ใช้

1. การเพิ่มเติมหรือใส่ข้อมูลคำอธิบายต่างๆ ให้กับสถาปัตยกรรมซอฟต์แวร์ที่ใช้งานอยู่ มีต้นทุนสูง เนื่องจากการที่จะต้องสูญเสียทรัพยากรในการวิเคราะห์และศึกษาข้อมูลใหม่อีกครั้ง
2. ปัจจัยของการออกแบบหรือปัจจัยของสภาพแวดล้อมที่มีอิทธิพลต่อการออกแบบสถาปัตยกรรมซอฟต์แวร์ อาจไม่ได้รับการอธิบายได้อย่างชัดเจนตั้งแต่เริ่มต้นโครงการ
3. เป้าหมายทางธุรกิจหรือข้อจำกัดต่างๆ ที่ได้กำหนดไว้ อาจได้รับการมองข้ามไป
4. ขาดการออกแบบสถาปัตยกรรมอย่างถูกต้อง ครบถ้วนตรงตามข้อกำหนดความต้องการ
5. การวิเคราะห์ผลดีผลเสียในการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์อาจถูกละเลย ส่งผลต่อการตัดสินใจ ทำให้เกิดการดำเนินการที่คลาดเคลื่อน
6. ผลกระทบต่อการเปลี่ยนแปลงความต้องการ หรือการปรับเปลี่ยนปัจจัยของสภาพแวดล้อมที่มีอิทธิพลกับการออกแบบสถาปัตยกรรมซอฟต์แวร์ อาจไม่ได้รับการให้ความสำคัญอย่างเหมาะสม

ประโยชน์ในการนำหลักของการระบุเหตุผลในการออกแบบไปประยุกต์ใช้ ยกตัวอย่างเช่น

1. ช่วยให้สถาปนิกและนักออกแบบเข้าใจเหตุผลของการออกแบบสถาปัตยกรรม
2. ช่วยให้สถาปนิกและนักออกแบบสามารถวิเคราะห์ผลกระทบต่อการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ได้
3. ช่วยให้สถาปนิกและนักออกแบบสามารถวิเคราะห์สาเหตุหลักของการออกแบบในลักษณะย้อนกลับโดยการตามรอยได้

4. สนับสนุนการตรวจสอบความครบถ้วนในการออกแบบและการบำรุงรักษา
5. การตรวจสอบลักษณะย้อนกลับเพื่อช่วยให้เข้าใจวิธีการและเหตุผลในการพัฒนาระบบ

จากการศึกษางานวิจัยดังกล่าวทำให้เข้าใจถึงที่มาและความสำคัญของการนำแนวคิดหลักของการระบุเหตุผลในการออกแบบสถาปัตยกรรมเป็นพื้นฐานไปประยุกต์ใช้งานในโครงการพัฒนาซอฟต์แวร์ โดยสามารถนำการวิเคราะห์ส่วนประกอบหรือสิ่งที่ควรคำนึงถึงในการออกแบบสถาปัตยกรรมซอฟต์แวร์ เพื่อที่จะสามารถระบุผลกระทบต่อการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์นำมาใช้เป็นข้อมูลตัวอย่างของเครื่องมือที่ทำการพัฒนาขึ้นในงานวิจัยต่อไปได้

บทที่ 3
การวิเคราะห์และการออกแบบ
การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์

ขั้นตอนการวิเคราะห์และออกแบบการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ในเอ็กซ์เอดีแอล ผู้วิจัยได้กำหนดการตรวจจับการเปลี่ยนแปลงเฉพาะส่วนประกอบพื้นฐานของสถาปัตยกรรมซอฟต์แวร์จากขอบเขตของงานวิจัยโดยทำการวิเคราะห์และออกแบบการจัดเก็บคำอธิบายข้อกำหนดความต้องการของโครงการ รวมถึงคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ ซึ่งอยู่บนหลักในการออกแบบสถาปัตยกรรมซอฟต์แวร์โดยใช้เหตุผลสำหรับการออกแบบเป็นพื้นฐาน [24] ทำให้สามารถตรวจจับผลกระทบ ที่อาจเกิดขึ้นได้ ในส่วนของการออกแบบวิธีการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ซึ่งผู้วิจัยให้ความสำคัญ ประกอบด้วย การอ่านโครงสร้างสถาปัตยกรรมซอฟต์แวร์ การตรวจจับความเปลี่ยนแปลงและผลกระทบที่อาจเกิดขึ้น รวมถึงแสดงผลรายงาน ซึ่งเชื่อมโยงไปยังคำอธิบายการออกแบบสถาปัตยกรรมซอฟต์แวร์ที่ได้กำหนดไว้ รายละเอียดดังต่อไปนี้

3.1 กำหนดส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ที่สนใจในการตรวจจับการเปลี่ยนแปลง

การตรวจจับการเปลี่ยนแปลงส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ มุ่งเน้นไปที่การตรวจจับรคุณลักษณะของส่วนประกอบพื้นฐานในภาษาเอ็กซ์เอดีแอล [18] ที่เป็นผลลัพธ์จากการออกแบบสถาปัตยกรรมซอฟต์แวร์โดยเครื่องมือ Archstudio 4 ซึ่งส่วนประกอบพื้นฐานดังกล่าวสามารถแยกได้จากแถบของข้อมูลของภาษา แบ่งออกเป็น 4 ส่วน ได้แก่ ส่วนประกอบ (Components) ตัวเชื่อม (Connectors) จุดเชื่อมต่อ (Interfaces) และเส้นเชื่อม (Links) โดยตัวอย่างแถบของข้อมูลในภาษาเอ็กซ์เอดีแอล แยกตามแต่ละส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ สามารถแสดงได้ดังรูปที่ 3.1 - 3.4

```
<instance:xArch xmlns:instance="http://www.ics.uci.edu/pub/arch/xArch/instance.xsd"
xmlns:hints3="http://www.ics.uci.edu/pub/arch/xArch/hints3.xsd"
xmlns:implementation="http://www.ics.uci.edu/pub/arch/xArch/implementation.xsd"
xmlns:lookupimplementation="http://www.ics.uci.edu/pub/arch/xArch/lookupimplementation.xsd"
xmlns:types="http://www.ics.uci.edu/pub/arch/xArch/types.xsd" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ics.uci.edu/pub/arch/xArch/lookupimplementation.xsd
http://www.isr.uci.edu/projects/xarchuci/ext/lookupimplementation.xsd
http://www.ics.uci.edu/pub/arch/xArch/implementation.xsd http://www.isr.uci.edu/projects/xarchuci/ext/implementation.xsd
http://www.ics.uci.edu/pub/arch/xArch/variants.xsd http://www.isr.uci.edu/projects/xarchuci/ext/variants.xsd
http://www.ics.uci.edu/pub/arch/xArch/hints3.xsd http://www.isr.uci.edu/projects/xarchuci/ext/hints3.xsd
http://www.ics.uci.edu/pub/arch/xArch/instance.xsd http://www.isr.uci.edu/projects/xarchuci/core/instance.xsd
http://www.ics.uci.edu/pub/arch/xArch/statecharts.xsd http://www.isr.uci.edu/projects/xarchuci/ext/statecharts.xsd
http://www.ics.uci.edu/pub/arch/xArch/activitydiagrams.xsd
http://www.isr.uci.edu/projects/xarchuci/ext/activitydiagrams.xsd http://www.ics.uci.edu/pub/arch/xArch/types.xsd
http://www.isr.uci.edu/projects/xarchuci/ext/types.xsd http://www.ics.uci.edu/pub/arch/xArch/changesets.xsd
http://www.isr.uci.edu/projects/xarchuci/ext/changesets.xsd">
    <types:archStructure types:id="archStructureffff9301-d19dd20b-3ef238a5-680d000a" xsi:type="types:ArchStructure">
        <types:description xsi:type="instance:Description">ABC_M1.2</types:description>
        <types:component types:id="componentffff9301-d19e11c5-71bf497e-680d0010" xsi:type="types:Component">
            <types:description xsi:type="instance:Description">Accounts</types:description>
            <types:interface types:id="interfaceffff9301-d1a321d6-f7a2f52b-680d031e" xsi:type="types:Interface">
                <types:description xsi:type="instance:Description">INF04</types:description>
                <types:direction xsi:type="instance:Direction">inout</types:direction>
                <types:type xlink:href="#5" xlink:type="simple" xsi:type="instance:XMLLink"/>
            </types:interface>
```

รูปที่ 3.1 ตัวอย่างแถบข้อมูลส่วนประกอบ

```
  </types:component>
  <types:connector types:id="connectorfffffffea-3b19029d-78cb4231-25ab0226" xsi:type="types:Connector">
      <types:description xsi:type="instance:Description">Connector1</types:description>
      <types:interface types:id="interfacefffffffea-3b193e08-7f32b974-25ab0263" xsi:type="types:Interface">
          <types:description xsi:type="instance:Description">INF-C1-1</types:description>
          <types:direction xsi:type="instance:Direction">inout</types:direction>
          <types:type xlink:href="#5" xlink:type="simple" xsi:type="instance:XMLLink"/>
      </types:interface>
      <types:interface types:id="interfacefffffffea-3b19490b-0ff5694c-25ab0282" xsi:type="types:Interface">
          <types:description xsi:type="instance:Description">INF-C1-2</types:description>
          <types:direction xsi:type="instance:Direction">inout</types:direction>
          <types:type xlink:href="#5" xlink:type="simple" xsi:type="instance:XMLLink"/>
      </types:interface>
      <types:type xlink:href="#4" xlink:type="simple" xsi:type="instance:XMLLink"/>
  </types:connector>
  <types:connector types:id="connectorfffffffea-3b19ae40-225fbf54-25ab030c" xsi:type="types:Connector">
      <types:description xsi:type="instance:Description">Connector3</types:description>
      <types:interface types:id="interfacefffffffea-3b19dacd-78de6602-25ab0363" xsi:type="types:Interface">
          <types:description xsi:type="instance:Description">INF-C3-2</types:description>
          <types:direction xsi:type="instance:Direction">inout</types:direction>
          <types:type xlink:href="#5" xlink:type="simple" xsi:type="instance:XMLLink"/>
      </types:interface>
```

รูปที่ 3.2 ตัวอย่างแถบข้อมูลตัวเชื่อม

```
<instance:xArch xmlns:instance="http://www.ics.uci.edu/pub/arch/xArch/instance.xsd"
xmlns:hints3="http://www.ics.uci.edu/pub/arch/xArch/hints3.xsd"
xmlns:implementation="http://www.ics.uci.edu/pub/arch/xArch/implementation.xsd"
xmlns:lookupimplementation="http://www.ics.uci.edu/pub/arch/xArch/lookupimplementation.xsd"
xmlns:types="http://www.ics.uci.edu/pub/arch/xArch/types.xsd" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ics.uci.edu/pub/arch/xArch/lookupimplementation.xsd
http://www.isr.uci.edu/projects/xarchuci/ext/lookupimplementation.xsd
http://www.ics.uci.edu/pub/arch/xArch/implementation.xsd http://www.isr.uci.edu/projects/xarchuci/ext/implementation.xsd
http://www.ics.uci.edu/pub/arch/xArch/variants.xsd http://www.isr.uci.edu/projects/xarchuci/ext/variants.xsd
http://www.ics.uci.edu/pub/arch/xArch/hints3.xsd http://www.isr.uci.edu/projects/xarchuci/ext/hints3.xsd
http://www.ics.uci.edu/pub/arch/xArch/instance.xsd http://www.isr.uci.edu/projects/xarchuci/core/instance.xsd
http://www.ics.uci.edu/pub/arch/xArch/statecharts.xsd http://www.isr.uci.edu/projects/xarchuci/ext/statecharts.xsd
http://www.ics.uci.edu/pub/arch/xArch/activitydiagrams.xsd
http://www.isr.uci.edu/projects/xarchuci/ext/activitydiagrams.xsd http://www.ics.uci.edu/pub/arch/xArch/types.xsd
http://www.isr.uci.edu/projects/xarchuci/ext/types.xsd http://www.ics.uci.edu/pub/arch/xArch/changesets.xsd
http://www.isr.uci.edu/projects/xarchuci/ext/changesets.xsd">
    <types:archStructure types:id="archStructureffff9301-d19dd20b-3ef238a5-680d000a" xsi:type="types:ArchStructure">
        <types:description xsi:type="instance:Description">ABC_M1.2</types:description>
        <types:component types:id="componentffff9301-d19e11c5-71bf497e-680d0010" xsi:type="types:Component">
            <types:description xsi:type="instance:Description">Accounts</types:description>
            <types:interface types:id="interfaceffff9301-d1a321d6-f7a2f52b-680d031e" xsi:type="types:Interface">
                <types:description xsi:type="instance:Description">INF04</types:description>
                <types:direction xsi:type="instance:Direction">inout</types:direction>
                <types:type xlink:href="#5" xlink:type="simple" xsi:type="instance:XMLLink"/>
            </types:interface>
```

รูปที่ 3.3 ตัวอย่างแถบข้อมูลจุดเชื่อมต่อ

```
<types:link types:id="linkfffffffea-3b1a188e-fe03fed8-25ab03cd" xsi:type="types:Link">
    <types:description xsi:type="instance:Description">Link1</types:description>
    <types:point xsi:type="instance:Point">
        <instance:anchorOnInterface xlink:href="#interfaceffff9301-d1a3620f-b829447c-680d0364" xlink:type="simple"
        xsi:type="instance:XMLLink"/>
    </types:point>
    <types:point xsi:type="instance:Point">
        <instance:anchorOnInterface xlink:href="#interfaceffffffea-3b193e08-7f32b974-25ab0263" xlink:type="simple"
        xsi:type="instance:XMLLink"/>
    </types:point>
</types:link>
<types:link types:id="linkfffffffea-3b1aeed0-06e0f3ff-25ab03f8" xsi:type="types:Link">
    <types:description xsi:type="instance:Description">Link6</types:description>
    <types:point xsi:type="instance:Point">
        <instance:anchorOnInterface xlink:href="#interfaceffffffea-3b19dacd-78de6602-25ab0363" xlink:type="simple"
        xsi:type="instance:XMLLink"/>
    </types:point>
    <types:point xsi:type="instance:Point">
        <instance:anchorOnInterface xlink:href="#interfaceffff9301-d1a3aa6e-45cc4f2a-680d03a4" xlink:type="simple"
        xsi:type="instance:XMLLink"/>
    </types:point>
</types:link>
```

รูปที่ 3.4 ตัวอย่างแถบข้อมูลเส้นเชื่อม

3.2 การจัดเก็บคำอธิบายข้อกำหนดความต้องการโครงการและการเชื่อมโยงกับคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์

ขั้นตอนนี้ ผู้วิจัยได้ออกแบบการจัดเก็บข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ ซึ่งผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์สามารถทำการบันทึกข้อมูลเบื้องต้นของโครงการ และรายการข้อกำหนดความต้องการซอฟต์แวร์ผ่านเครื่องมือที่พัฒนาขึ้น โดยข้อมูลที่ได้รับการบันทึกจะถูกจัดเก็บเป็นแฟ้มข้อมูลประเภทซีเอสวี โดยมีรายละเอียด ดังนี้

[[projectId],[projectCode],[projectName],[projectDescription],[requirementId], [requirementCode],[requirementName],[requirementDescription],[requirementStatus], [requirementReporter],[requirementAssignedTo],[requirementSubmittedDate], [requirementUpdatedDate]]*

รายละเอียดข้อมูลที่บันทึก มีดังต่อไปนี้

[projectId]	แสดงรหัสระบุโครงการซอฟต์แวร์
[projectCode]	แสดงรหัสโครงการซอฟต์แวร์
[projectName]	แสดงชื่อโครงการซอฟต์แวร์
[projectDescription]	แสดงคำอธิบายโครงการซอฟต์แวร์
[requirementId]	แสดงรหัสระบุความต้องการ
[requirementCode]	แสดงรหัสความต้องการ
[requirementName]	แสดงชื่อความต้องการ
[requirementDescription]	แสดงคำอธิบายความต้องการ

[requirementStatus]	แสดงสถานะความต้องการ
[requirementReporter]	แสดงชื่อผู้รายงานความต้องการ
[requirementAssignedToi]	แสดงชื่อผู้รับผิดชอบต่อความต้องการ
[requirementSubmittedDate]	แสดงวันที่สร้างความต้องการ
[requirementUpdatedDate]	แสดงวันที่ปรับปรุงข้อมูลความต้องการ
	เครื่องหมาย * แสดงการระบุข้อมูลได้มากกว่า 1 ค่า โดยที่ รหัสระบุความต้องการและ สามารถระบุอ้างอิงให้กับระบุโครงการซอฟต์แวร์ได้มากกว่า 1 ค่า

ตัวอย่างการจัดเก็บข้อมูลคำอธิบายข้อกำหนดความต้องการ สามารถแสดงได้ดังรูปที่ 3.5

```
projectId,projectCode,projectName,projectDescription,i
7946384435390655260,ABC-M1,Develops software solutions
7946384435390655260,ABC-M1,Develops software solutions
7946384435390655260,ABC-M1,Develops software solutions
7946384435390655260,ABC-M1,Develops software solutions
7946384435390655260,ABC-M1,Develops software solutions
7946384435390655260,ABC-M1,Develops software solutions
7946384435390655260,ABC-M1,Develops software solutions
7946384435390655260,ABC-M1,Develops software solutions
```

รูปที่ 3.5 ตัวอย่างรูปแบบแฟ้มข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ

นอกจากนี้ ผู้วิจัยได้ออกแบบการจัดเก็บข้อมูลคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ โดยบันทึกข้อมูลทั้งหมด ในลักษณะคำอธิบายแนบไปกับแฟ้มข้อมูลในภาษาเอ็กซ์เอดีแอล ดังนี้

<! -- [projectCode] | [[componentID], [requirementsID]*, [qualityAttributesID]*]* -->

รายละเอียดข้อมูลที่บันทึก มีดังต่อไปนี้

[projectCode]	แสดงรหัสโครงการซอฟต์แวร์
[componentID]	แสดงรหัสส่วนประกอบ
[requirementsID]	แสดงรหัสความต้องการ
[qualityAttributesID]	แสดงรหัสคุณลักษณะเชิงคุณภาพ

เครื่องหมาย *	แสดงการระบุข้อมูลได้มากกว่า 1 ค่า โดยที่ รหัสความต้องการและรหัสคุณลักษณะเชิงคุณภาพ สามารถระบุอ้างอิงให้กับรหัสส่วนประกอบได้มากกว่า 1 ค่า

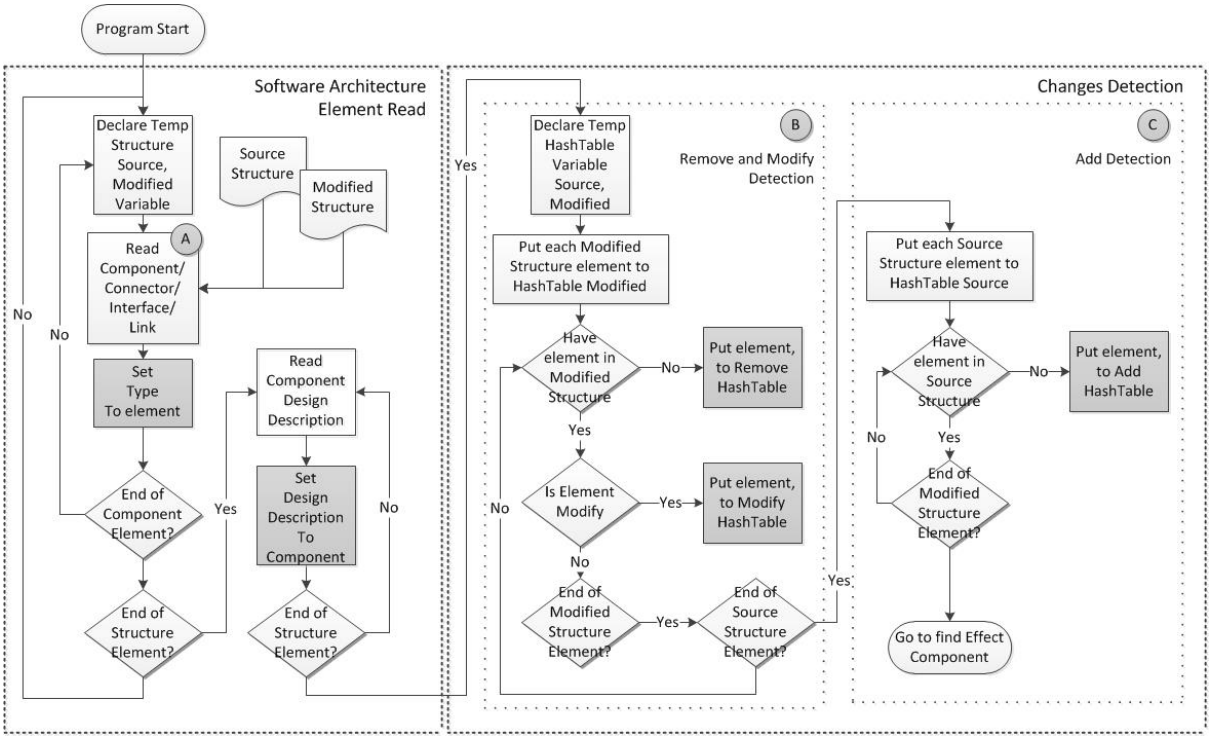
ตัวอย่างการจัดเก็บคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ สามารถแสดงได้ดังรูปที่ 3.6

```
        <types:interfaceType types:id="interfaceTypeffff9301-0739312b-f14e22
        xsi:type="implementation:InterfaceTypeImpl">
            <types:description xsi:type="instance:Description">SecureInterfa
            <implementation:implementation xsi:type="lookupimplementation:Lo
                <lookupimplementation:name
                xsi:type="lookupimplementation:LookupName">com.abc.SecureInt
            </implementation:implementation>
        </types:interfaceType>
    </types:archTypes>
</instance:xArch>
<!--
ABC-CODE-M01|componentffff9301-d19e11c5-71bf497e-680d0010,1,QA2,QA4,QA5,QA6|
e,2|componentffff9301-d19fe14b-f0db263e-680d00bf,3,QA1|componentffff9301-d1a
9301-d1a178ae-256cda51-680d01a4,5,QA3|componentffff9301-d1a1cf84-651bb10e-68
665c5-680d0205,7,QA5|componentffff9301-d1a27739-a14f1a19-680d025e,8,QA6 -->
```

รูปที่ 3.6 ตัวอย่างการจัดเก็บคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์

3.3 ออกแบบวิธีการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์

ในขั้นตอนของการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ แบ่งการทำงานออกเป็น 3 ส่วน ได้แก่ การอ่านโครงสร้างสถาปัตยกรรมซอฟต์แวร์ต้นฉบับ รวมถึงสถาปัตยกรรมซอฟต์แวร์ที่ถูกแก้ไข และการตรวจจับความเปลี่ยนแปลง การตรวจจับผลกระทบจากการเปลี่ยนแปลง และการแสดงผลรายงาน ซึ่งได้แสดงขั้นตอนการทำงาน รวมถึงเงื่อนไขการตรวจจับ แสดงไว้ในรูปที่ 3.7 - 3.9



รูปที่ 3.7 การอ่านโครงสร้างสถาปัตยกรรมซอฟต์แวร์ต้นฉบับและสถาปัตยกรรมซอฟต์แวร์ที่ถูกแก้ไข และการตรวจจับความเปลี่ยนแปลง

1) **การอ่านส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ และการตรวจจับความเปลี่ยนแปลง** (Software Architecture Element Read/Changes Detection) ขั้นตอนแรกของการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ จะเริ่มต้นจากการอ่านโครงสร้างของแฟ้มข้อมูลนำเข้าสถาปัตยกรรมซอฟต์แวร์ต้นฉบับ และสถาปัตยกรรมซอฟต์แวร์ที่ถูกแก้ไขในขั้นตอน A รวมถึงคำอธิบายในการออกแบบของส่วนประกอบ (ถ้ามี) โดยนำข้อมูลรายละเอียดต่างๆ จัดเก็บไปยังจาวาอ็อบเจ็กต์ที่ได้ออกแบบให้สอดคล้องกับขอบเขตของการตรวจจับผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ โดยรายละเอียดแสดงได้ดังแผนภาพคลาสในส่วนของการออกแบบและการพัฒนาเครื่องมือในบทที่ 4

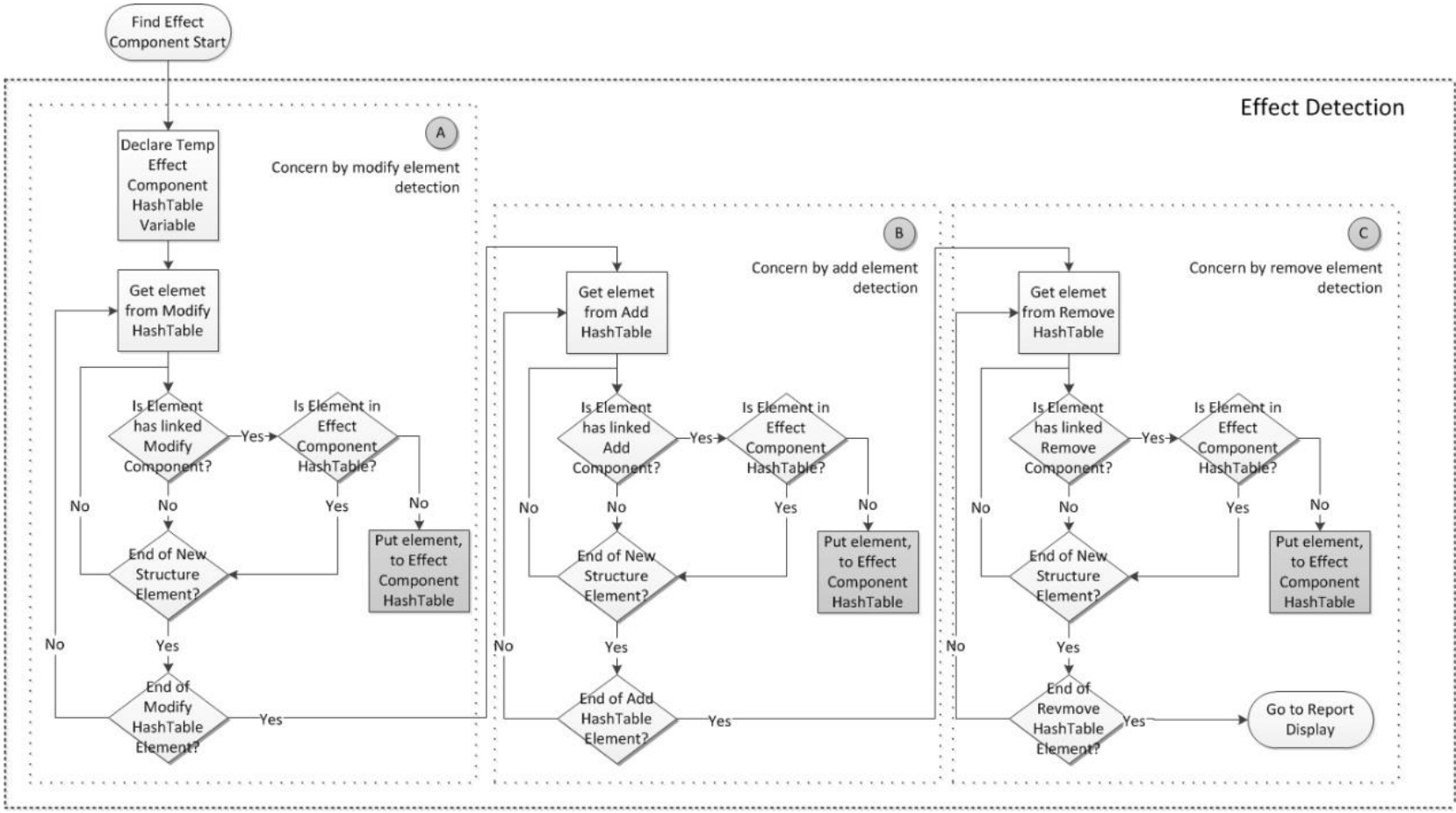
หลังจากขั้นตอนการอ่านโครงสร้างส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์เสร็จสิ้น จากนั้นจะเข้าสู่ขั้นตอนการเปรียบเทียบความแตกต่างระหว่างข้อมูลจาวาอ็อบเจ็กต์ของสถาปัตยกรรมซอฟต์แวร์ต้นฉบับ และสถาปัตยกรรมซอฟต์แวร์ที่ถูกแก้ไข ซึ่งแบ่งออกเป็น 2 ส่วน ได้แก่

- **การตรวจจับการลบออกและการแก้ไข** (Remove and Modify Detection) ในขั้นตอน B ใช้วิธีการตรวจจับการลบออกโดยการนำข้อมูลรหัสระบุตัวตน (Unique identifier) ของส่วนประกอบภายในจาวาอ็อบเจ็กต์ของสถาปัตยกรรมต้นฉบับ ค้นหากับข้อมูลรหัสระบุตัวตนของส่วนประกอบภายในจาวาอ็อบเจ็กต์ของสถาปัตยกรรมที่ถูกแก้ไข ในกรณีที่ไม่พบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ต้นฉบับ จึงทำการบันทึกส่วนประกอบที่ถูกลบออกไปยังตัวแปรชั่วคราวในลักษณะของตารางแฮช

นอกจากนี้ วิธีการตรวจจับการแก้ไขของส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ ใช้วิธีการค้นหาข้อมูลภายในจาวาอ็อบเจ็กต์ของสถาปัตยกรรมต้นฉบับ ที่มีอยู่ในจาวาอ็อบเจ็กต์ของสถาปัตยกรรมที่ถูกแก้ไข จากนั้นจึงใช้การเทียบคุณสมบัติของส่วนประกอบตามขอบเขตของเครื่องมือที่ได้กำหนดไว้ กรณีที่คุณสมบัติของส่วนประกอบมีการปรับเปลี่ยนไปจากค่าที่ถูกจัดเก็บเดิม จึงทำการบันทึกส่วนประกอบที่ถูกแก้ไขไปยังตัวแปรชั่วคราวในลักษณะของตารางแฮช

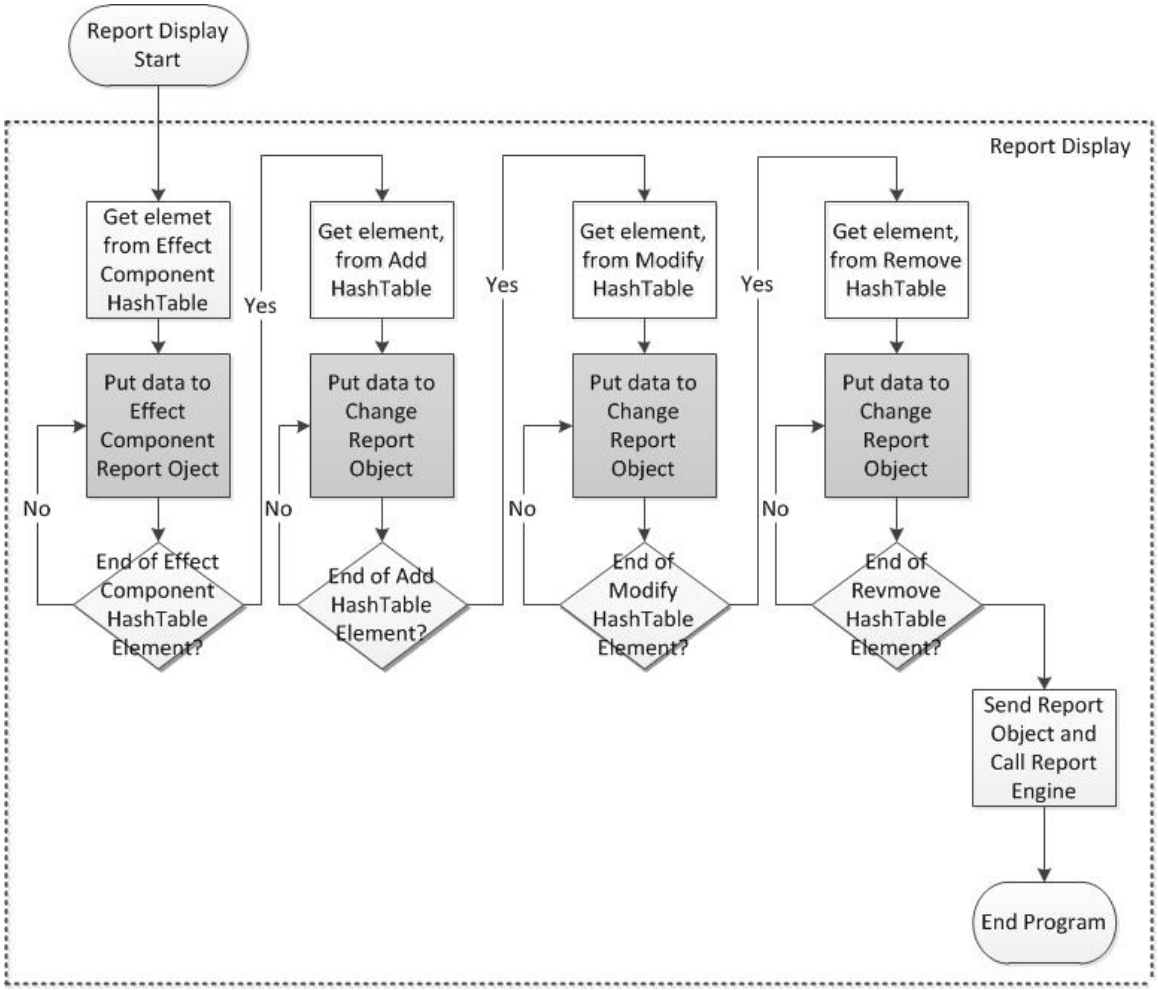
- **การตรวจจับการเพิ่ม** (Add Detection) ในขั้นตอน C ใช้วิธีการตรวจจับการเพิ่มโดยการนำข้อมูลรหัสระบุตัวตนของส่วนประกอบภายในจาวาอ็อบเจ็กต์ของสถาปัตยกรรมที่ถูกแก้ไข ค้นหากับข้อมูลรหัสระบุตัวตนของส่วนประกอบภายในจาวาอ็อบเจ็กต์ของสถาปัตยกรรมต้นฉบับ ในกรณีที่ไม่พบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ที่ถูกแก้ไข จึงทำการบันทึกส่วนประกอบที่ถูกออกแบบเพิ่มไปยังตัวแปรชั่วคราวในลักษณะของตารางแฮช

โดยข้อมูลของส่วนประกอบที่ได้รับการเปลี่ยนแปลงและจัดเก็บไว้ที่ตัวแปรชั่วคราวในลักษณะของตารางแฮชข้างต้น จะถูกนำไปใช้กับขั้นตอนการตรวจจับผลกระทบ ในขั้นตอนต่อไป ดังแสดงในรูปที่ 3.8



รูปที่ 3.8 การตรวจจับผลกระทบ

2) **การตรวจจับผลกระทบ** (Effect Detection) หลังจากผ่านขั้นตอนการอ่านโครงสร้างสถาปัตยกรรมซอฟต์แวร์ต้นฉบับ และสถาปัตยกรรมซอฟต์แวร์ที่ถูกแก้ไข และการตรวจจับความเปลี่ยนแปลงแล้ว ข้อมูลที่ได้นำมาใช้ในขั้นตอนนี้ คือ ตัวแปรชั่วคราวในลักษณะของตารางแฮช ซึ่งการตรวจจับส่วนประกอบที่ได้รับผลกระทบ ใช้วิธีการค้นหาส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ที่ถูกแก้ไข และที่มีเส้นเชื่อมสัมพันธ์กับส่วนประกอบที่ได้รับ การแก้ไขในขั้นตอน A การเพิ่มในขั้นตอน B และการลบออกในขั้นตอน C จากข้อมูลส่วนประกอบที่ตรวจจับได้ในขั้นตอนแรก เมื่อตรวจจับพบส่วนประกอบที่ได้รับผลกระทบ จึงทำการบันทึกส่วนประกอบดังกล่าวไปยังตัวแปรชั่วคราวในลักษณะของตารางแฮช เพื่อนำไปใช้ส่วนของการแสดงผลรายงาน ในขั้นตอนต่อไป ดังแสดงได้ในรูปที่ 3.9



รูปที่ 3.9 การแสดงผลรายงาน

3) **การแสดงผลรายงาน** (Report Display) เป็นขั้นตอนในการนำข้อมูลผลลัพธ์ที่ได้รับการประมวลผลจากการตรวจจับความเปลี่ยนแปลงและผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ในขั้นตอนก่อนหน้า ซึ่งเป็นการจัดเก็บข้อมูลในลักษณะของตารางแฮชแยกตามลักษณะของรายงานที่ใช้ในการแสดงผลลัพธ์ นำมาจัดรูปแบบ จากนั้นจัดเก็บไปยังจาวาอ็อบเจ็กต์ เพื่อการแสดงรายงาน และส่งต่อข้อมูลที่ได้รับการจัดรูปแบบเรียบร้อยแล้วสำหรับเรียกใช้การทำงานของโปรแกรมสำหรับแสดงผลลัพธ์ รายละเอียดข้อมูลที่แสดงในรายงานผลลัพธ์ของเครื่องมือที่พัฒนาขึ้น ได้แสดงตัวอย่างไว้ในบทที่ 5

3.4 ผลกระทบที่ได้จากการตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์

สำหรับงานวิจัยนี้ข้อมูลผลกระทบจากการเปลี่ยนแปงสถาปัตยกรรมซอฟต์แวร์ สามารถแบ่งออกได้เป็น 3 ส่วน ได้แก่ ผลกระทบต่อข้อกำหนดความต้องการโครงการ ผลกระทบต่อคุณลักษณะเชิงคุณภาพของซอฟต์แวร์ [15] และผลกระทบต่อชุดคำสั่งที่เกี่ยวข้องกับการออกแบบส่วนประกอบ

ทั้งนี้ การแสดงผลกระทบในส่วนที่หนึ่งผลกระทบต่อข้อกำหนดความต้องการโครงการ และส่วนที่สองผลกระทบต่อคุณลักษณะเชิงคุณภาพ ข้อมูลที่นำมาใช้แสดงผลมาจากการที่นักออกแบบสถาปัตยกรรมซอฟต์แวร์ ได้กำหนดคำอธิบายการออกให้กับส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ โดยเครื่องมือที่พัฒนาขึ้น รายละเอียดในส่วนของการกำหนดอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ ได้แสดงไว้ในบทที่ 4 การออกแบบและการพัฒนาเครื่องมือ

ส่วนการแสดงผลในส่วนที่สาม ผลกระทบต่อชุดคำสั่งที่เกี่ยวข้องกับการออกแบบส่วนประกอบ ข้อมูลที่นำมาแสดงผลมาจากการกำหนดชุดคำสั่งที่เกี่ยวข้องตามประเภทของส่วนประกอบของเครื่องมือ Archstudio 4 ที่ใช้ในการออกแบบสถาปัตยกรรมซอฟต์แวร์ โดยข้อมูลจะถูกจัดเก็บเป็นคุณลักษณะของส่วนประกอบในขั้นตอนของการอ่านส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ รายละเอียดดังรูปที่ 3.10

```
                </hints3:hintedElement>
            </hints3:hintedElement>
        </hints3:hintedElement>
    </hints3:renderingHints3>
<types:archTypes xsi:type="types:ArchTypes">
        <types:componentType types:id="1" xsi:type="implementation:VariantComponentTypeImpl">
            <types:description xsi:type="instance:Description">FinancialComponent</types:description>
            <implementation:implementation xsi:type="lookupimplementation:LookupImplementation">
                <lookupimplementation:name
                xsi:type="lookupimplementation:LookupName">com.abc.financialImpl</lookupimplementation:name>
            </implementation:implementation>
        </types:componentType>
```

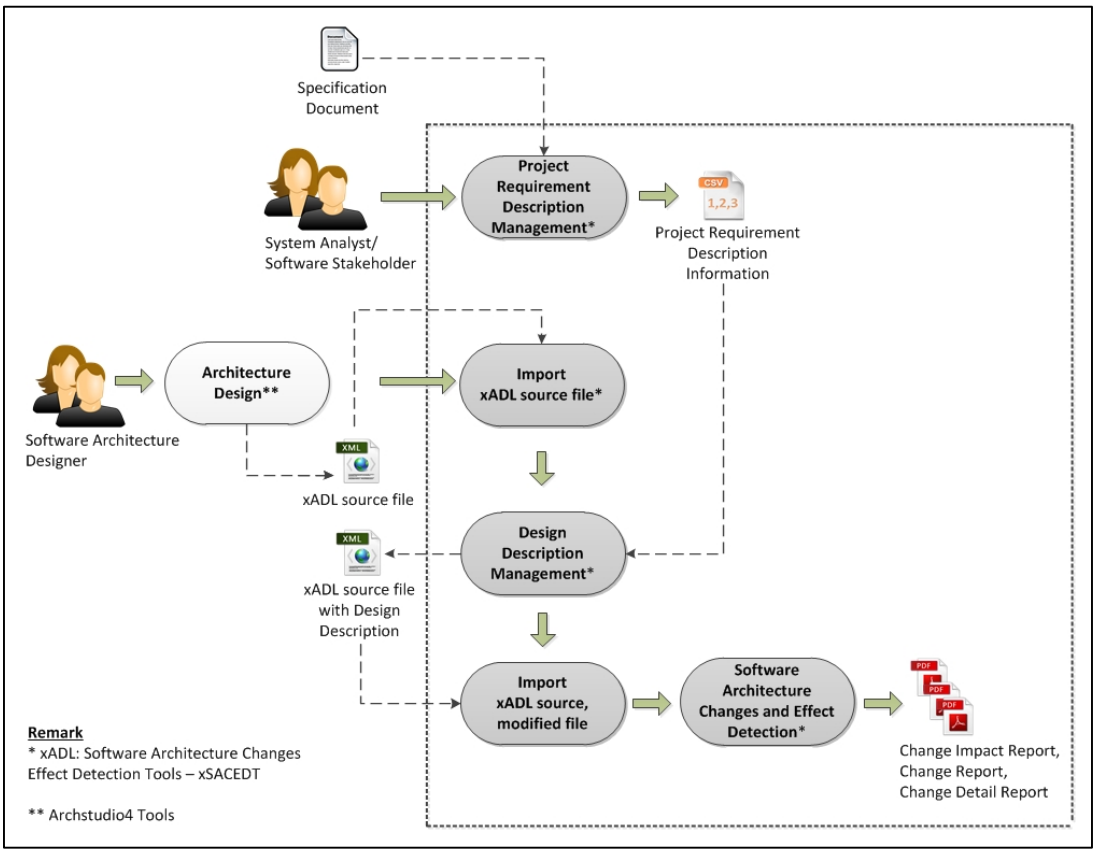
รูปที่ 3.10 ตัวอย่างแถบข้อมูลคำอธิบายชุดคำสั่งที่เกี่ยวข้องกับประเภทส่วนประกอบ

บทที่ 4
การออกแบบและการพัฒนาเครื่องมือ

รายละเอียดเนื้อหาที่กล่าวในบทนี้จะอธิบายถึงการออกแบบ การพัฒนาเครื่องมือ รวมถึงสภาพแวดล้อมที่ใช้สำหรับพัฒนาเครื่องมือ และโครงสร้างของส่วนต่อประสานกับผู้ใช้งานของเครื่องมือที่พัฒนา เพื่อสนับสนุนการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์

4.1 โครงสร้างเครื่องมือการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ในเอ็กซ์เอดีแอล

ภาพรวมโครงสร้างเครื่องมือการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ในเอ็กซ์เอดีแอลแสดงได้ดังรูปที่ 4.1



รูปที่ 4.1 ภาพรวมโครงสร้างเครื่องมือการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ในเอ็กซ์เอดีแอล

1) **การจัดการคำอธิบายข้อกำหนดความต้องการโครงการ** (Project Requirement Description Management) ในขั้นตอนแรกหลังจากเริ่มต้นกระบวนการพัฒนาโครงการซอฟต์แวร์ นักวิเคราะห์ระบบหรือผู้ที่เกี่ยวข้องกับโครงการซอฟต์แวร์ นำข้อมูลที่เกี่ยวข้องกับโครงการเบื้องต้นมาบันทึกผ่านเครื่องมือ โดยข้อมูลประกอบไปด้วย ชื่อโครงการ รหัส คำอธิบาย รวมไปถึงรายการข้อกำหนดความต้องการที่เกี่ยวข้องกับโครงการ ซึ่งมีข้อมูลประกอบด้วย ชื่อความต้องการ คำอธิบาย สถานะ ชื่อผู้รายงาน และชื่อผู้รับผิดชอบต่อความต้องการ ข้อมูลที่ได้รับการบันทึกจะถูกจัดเก็บในรูปแบบแฟ้มข้อมูลประเภทซีเอสวี เพื่อนำข้อมูลไปใช้ในขั้นตอนการระบุคำอธิบายการออกแบบต่อไป ทั้งนี้ นักวิเคราะห์ระบบหรือผู้ที่เกี่ยวข้องกับโครงการซอฟต์แวร์ยังสามารถนำเข้า (Import) และส่งออก (Export) ข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการที่ได้ทำการบันทึกไว้ เพื่อนำข้อมูลไปใช้งานในขั้นตอนอื่นๆ ของกระบวนการพัฒนาซอฟต์แวร์ต่อไปได้อีกด้วย

2) **การออกแบบสถาปัตยกรรมซอฟต์แวร์** (Architecture Design) ในส่วนของการออกแบบสถาปัตยกรรมซอฟต์แวร์ นักออกแบบสถาปัตยกรรมซอฟต์แวร์จะทำการออกแบบแบบจำลองสถาปัตยกรรมซอฟต์แวร์ผ่านเครื่องมือ Archstudio 4 ซึ่งแบบจำลองสถาปัตยกรรมซอฟต์แวร์ที่ออกแบบจะมีความสอดคล้องกับข้อกำหนดความต้องการโครงการซอฟต์แวร์ รวมถึงคุณลักษณะเชิงคุณภาพ โดยผลลัพธ์ที่ได้จากการออกแบบ คือ แบบจำลองสถาปัตยกรรมซอฟต์แวร์ที่จัดเก็บอยู่ในรูปแบบแฟ้มข้อมูลภาษาเอ็กซ์เอดีแอล เวอร์ชัน 2.0

3) **นำเข้าแฟ้มข้อมูลเอ็กซ์เอดีแอล** (Import xADL source file) หลังจากนักออกแบบสถาปัตยกรรมซอฟต์แวร์ได้ทำการออกแบบแบบจำลองสถาปัตยกรรมซอฟต์แวร์เสร็จสิ้น นักออกแบบจะดำเนินการกำหนดคำอธิบายการออกแบบส่วนประกอบต่างๆ ของสถาปัตยกรรมซอฟต์แวร์ โดยการนำเข้าแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์ที่มีรูปแบบของภาษาเอ็กซ์เอดีแอลผ่านเครื่องมือที่พัฒนาขึ้นในขั้นตอนนี้

4) **การจัดการคำอธิบายการออกแบบ** (Design Description Management) นักออกแบบสถาปัตยกรรมซอฟต์แวร์ สามารถจัดการคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ ซึ่งใช้เป็นข้อมูลเบื้องต้นในการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ ทั้งนี้ ข้อมูลที่ผู้ใช้งานสามารถกำหนดให้กับส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ได้ แบ่งออกเป็นสองส่วน คือ รายการข้อกำหนดความต้องการที่เกี่ยวข้อง

กับการออกแบบส่วนประกอบ (ผู้ใช้งานเลือกข้อมูลจากส่วนของการจัดการคำอธิบายข้อกำหนดความต้องการโครงการในข้อที่หนึ่ง และคุณลักษณะเชิงคุณภาพที่เกี่ยวข้องกับการออกแบบส่วนประกอบ

5) **นำเข้าแฟ้มข้อมูลเอ็กซ์เอดีแอล ต้นแบบและ แฟ้มข้อมูลที่ถูกแก้ไข** (Import xADL source, modified file) เมื่อนักวิเคราะห์ระบบหรือนักออกแบบสถาปัตยกรรมซอฟต์แวร์ ต้องการทราบถึงผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ ผู้ใช้งานทำการนำเข้าแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์ก่อนและหลังการแก้ไข ผ่านเครื่องมือที่พัฒนาขึ้นในขั้นตอนนี้

6) **การตรวจจับความเปลี่ยนแปลงและผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์** (Software Architecture Changes and Effect Detection) เป็นส่วนที่ทำการเปรียบเทียบความแตกต่างระหว่างแบบจำลองสถาปัตยกรรมซอฟต์แวร์ก่อนและหลังการแก้ไข จากส่วนประกอบที่ได้รับการเพิ่ม การลบออก หรือการแก้ไขคุณลักษณะตามขอบเขตที่เครื่องมือสามารถตรวจจับได้ โดยตรวจจับส่วนประกอบที่ได้รับผลกระทบจากการเชื่อมต่อกับส่วนประกอบที่มีการเปลี่ยนแปลงคุณลักษณะ ทั้งนี้ การแสดงผลลัพธ์ของเครื่องมือที่พัฒนาขึ้นแสดงผลในรูปแบบรายงาน แบ่งออกเป็นสามส่วน คือ รายงานผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ ซึ่งรวบรวมข้อมูลรายละเอียดข้อกำหนดความต้องการซอฟต์แวร์ คุณลักษณะเชิงคุณภาพ หรือชุดคำสั่งใดๆ ที่ได้รับผลกระทบจากการเปลี่ยนแปลง รายงานสรุปผลความเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ และรายงานแสดงรายละเอียดการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์

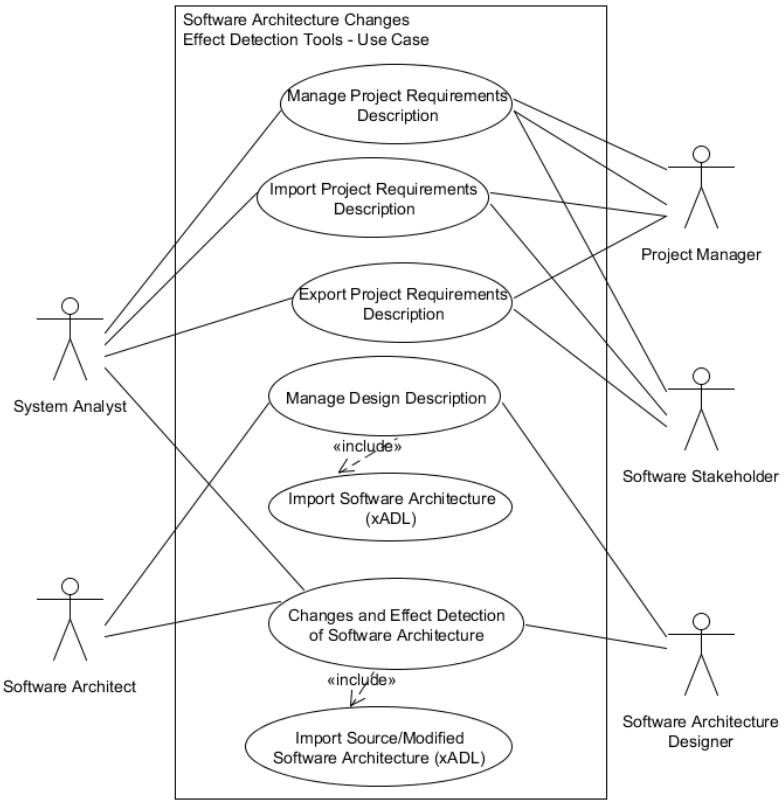
4.2 พัฒนาเครื่องมือที่ใช้ในการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์

ในการพัฒนาเครื่องมือที่ใช้ในการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ได้ออกแบบการพัฒนาเครื่องมือโดยอธิบายการออกแบบด้วยแผนภาพในภาษา ยูเอ็มแอลประกอบไปด้วย แผนภาพยูสเคส แผนภาพคลาส และแผนภาพกิจกรรม โดยมีการอธิบายถึง

โครงสร้างของการจัดเก็บข้อมูลซึ่งมีความสัมพันธ์กันระหว่างคุณลักษณะของโครงสร้างข้อมูลและแผนภาพ มีรายละเอียดดังนี้

4.2.1 แผนภาพยูสเคส

เครื่องมือสำหรับการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ได้ถูกอธิบายถึงพื้นฐานของเครื่องมือในมุมมองของการใช้งานด้วยแผนภาพยูสเคสดังรูปที่ 4.2



รูปที่ 4.2 แผนภาพยูสเคสของเครื่องมือที่ใช้สำหรับการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์

เครื่องมือสำหรับการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ประกอบได้ด้วย 7 ยูสเคส ได้แก่ 1. ยูสเคสจัดการคำอธิบายข้อกำหนดความต้องการโครงการ 2. ยูสเคสนำเข้าคำอธิบายข้อกำหนดความต้องการโครงการ 3. ยูสเคสส่งออกคำอธิบายข้อกำหนดความต้องการโครงการ 4 ยูสเคสจัดการคำอธิบายการออกแบบส่วนประกอบ 5. ยูสเคสนำเข้าแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์ 6 ยูสเคสตรวจจับความเปลี่ยนแปลงและผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์และ 7. ยูสเคสนำเข้าสถาปัตยกรรมซอฟต์แวร์ต้นฉบับและสถาปัตยกรรมซอฟต์แวร์ที่ถูกแก้ไข โดยในแต่ละยูสเคสได้แสดงรายละเอียดไว้ดังตารางที่ 4.1 - 4.7

ตารางที่ 4.1 คำอธิบายยูสเคสจัดการคำอธิบายข้อกำหนดความต้องการโครงการ

ชื่อยูสเคส	จัดการคำอธิบายข้อกำหนดความต้องการโครงการ
ผู้กระทำ	นักวิเคราะห์ระบบ,ผู้จัดการโครงการ ผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์
รายละเอียดยูสเคส	การเพิ่ม แก้ไข รายละเอียดเบื้องต้นเกี่ยวกับโครงการพัฒนาซอฟต์แวร์ ยกตัวอย่างเช่น รหัสโครงการ, ชื่อโครงการ, คำอธิบายโครงการ รวมถึงรายการข้อกำหนดความต้องการที่เกี่ยวข้องกับโครงการ ในรูปแบบแฟ้มข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ เพื่อนำข้อมูลที่บันทึกไปใช้ในขั้นตอนการระบุคำอธิบายการออกแบบ โดยแฟ้มข้อมูลดังกล่าวจะถูกจัดเก็บเป็นแฟ้มข้อมูลประเภทซีวีเอส
ขั้นตอน	1. ผู้ใช้งานเข้าสู่หน้าจอการจัดการคำอธิบายข้อกำหนดความต้องการโครงการ 2. ผู้ใช้งานเลือกเพิ่ม หรือแก้ไขรายละเอียดโครงการ 3. ผู้ใช้งานเลือกเพิ่ม, ลบ หรือแก้ไขคำอธิบายข้อกำหนดความต้องการโครงการ 4. ผู้ใช้งานเลือกบันทึกข้อมูล
เงื่อนไขภายหลัง	แสดงคำอธิบายข้อกำหนดความต้องการโครงการแยกตามรหัสโครงการ

ตารางที่ 4.2 คำอธิบายยูสเคสนำเข้าคำอธิบายข้อกำหนดความต้องการโครงการ

ชื่อยูสเคส	นำเข้าคำอธิบายข้อกำหนดความต้องการโครงการ
ผู้กระทำ	นักวิเคราะห์ระบบ,ผู้จัดการโครงการ ผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์
รายละเอียดยูสเคส	การนำเข้า คำอธิบายข้อกำหนดโครงการ ในรูปแบบแฟ้มข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ เพื่อนำข้อมูลที่นำเข้าไปใช้ในขั้นตอนการระบุคำอธิบายการออกแบบ โดยแฟ้มข้อมูลดังกล่าวจะถูกจัดเก็บเป็นแฟ้มข้อมูลประเภทซีวีเอส
เงื่อนไขก่อนหน้า	การนำเข้าข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ แฟ้มข้อมูลจะต้องมีรูปแบบตามที่กำหนดไว้

ตารางที่ 4.2 คำอธิบายยูสเคสนำเข้าคำอธิบายข้อกำหนดความต้องการโครงการ (ต่อ)

ขั้นตอน	1. ผู้ใช้งานเข้าสู่หน้าจอการจัดการคำอธิบายข้อกำหนดความต้องการโครงการ 2. ผู้ใช้งานเลือกนำเข้า 3. ผู้ใช้งานเลือกแฟ้มข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ 4. ผู้ใช้งานเลือกยืนยันการนำเข้าข้อมูล
เงื่อนไขภายหลัง	-

ตารางที่ 4.3 คำอธิบายยูสเคสส่งออกคำอธิบายข้อกำหนดความต้องการโครงการ

ชื่อยูสเคส	ส่งออกคำอธิบายข้อกำหนดความต้องการโครงการ
ผู้กระทำ	นักวิเคราะห์ระบบ,ผู้จัดการโครงการ ผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์
รายละเอียดยูสเคส	การส่งออกคำอธิบายข้อกำหนดโครงการ ในรูปแบบแฟ้มข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ เพื่อนำข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการที่ได้บันทึกไว้ไปใช้งานกับกระบวนการพัฒนาซอฟต์แวร์ในภายหลังได้ โดยแฟ้มข้อมูลดังกล่าวจะถูกจัดเก็บเป็นแฟ้มข้อมูลประเภทซีวีเอส
เงื่อนไขก่อนหน้า	-
ขั้นตอน	1. ผู้ใช้งานเข้าสู่หน้าจอการจัดการคำอธิบายข้อกำหนดความต้องการโครงการ 2. ผู้ใช้งานเลือกข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ 3. ผู้ใช้งานเลือกส่งออก 4. ผู้ใช้งานเลือกที่อยู่สำหรับการบันทึกข้อมูล 5. ผู้ใช้งานเลือกยืนยันการส่งออกข้อมูล
เงื่อนไขภายหลัง	แฟ้มข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ จะถูกบันทึกในรูปแบบตามที่กำหนดไว้

ตารางที่ 4.4 คำอธิบายยูสเคสจัดการคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์

ชื่อยูสเคส	จัดการคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์
ผู้กระทำ	สถาปนิกซอฟต์แวร์, นักออกแบบสถาปัตยกรรมซอฟต์แวร์

ตารางที่ 4.4 คำอธิบายยูสเคสจัดการคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ (ต่อ)

รายละเอียดยูสเคส	การจัดการคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ ใช้เป็นข้อมูลเบื้องต้นในการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ ข้อมูลที่ผู้ใช้งานสามารถกำหนดให้กับส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ได้ แบ่งออกเป็นสองส่วน คือ 1. รายการความต้องการที่เกี่ยวข้องกับการออกแบบส่วนประกอบ 2. คุณลักษณะเชิงคุณภาพที่เกี่ยวข้องกับการออกแบบส่วนประกอบ
เงื่อนไขก่อนหน้า	แฟ้มข้อมูลสถาปัตยกรรมถูกนำเข้าและพบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ที่ได้รับการออกแบบ
ขั้นตอน	1. ผู้ใช้งานเข้าสู่หน้าจอการระบุคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ 2. ผู้ใช้งานเลือกข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ 3. ผู้ใช้งานเลือกเพิ่ม, ลบ หรือแก้ไขคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ 4. ผู้ใช้งานเลือกยืนยันการบันทึกข้อมูล
เงื่อนไขภายหลัง	คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ถูกบันทึกในลักษณะคำอธิบายแนบไปกับแฟ้มข้อมูลในภาษาเอ็กซ์เอดีแอล

ตารางที่ 4.5 คำอธิบายยูสเคสนำเข้าแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์

ชื่อยูสเคส	นำเข้าแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์
ผู้กระทำ	สถาปนิกซอฟต์แวร์, นักออกแบบสถาปัตยกรรมซอฟต์แวร์
รายละเอียดยูสเคส	การนำเข้าแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์ ซึ่งกำหนดให้ผู้ใช้งานเลือกข้อมูลสถาปัตยกรรมซอฟต์แวร์ที่ได้ออกแบบ เพื่อการกำหนดคำอธิบายการออกแบบให้กับส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์
เงื่อนไขก่อนหน้า	แฟ้มข้อมูลนำเข้าที่ใช้กำหนดคำอธิบายการออกแบบส่วนประกอบจะต้องมีรูปแบบเป็นภาษาเอ็กซ์เอดีแอล เวอร์ชันสอง

ตารางที่ 4.5 คำอธิบายยูสเคสนำเข้าแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์ (ต่อ)

ขั้นตอน	1. ผู้ใช้งานเข้าสู่หน้าจอการระบุคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ 2. ผู้ใช้งานเลือกนำเข้าแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์ 3. ผู้ใช้งานเลือกที่อยู่ของแฟ้มข้อมูล 4. ผู้ใช้งานเลือกยืนยันการนำเข้าข้อมูล
เงื่อนไขภายหลัง	-

ตารางที่ 4.6 คำอธิบายยูสเคสตรวจจับความเปลี่ยนแปลงและผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์

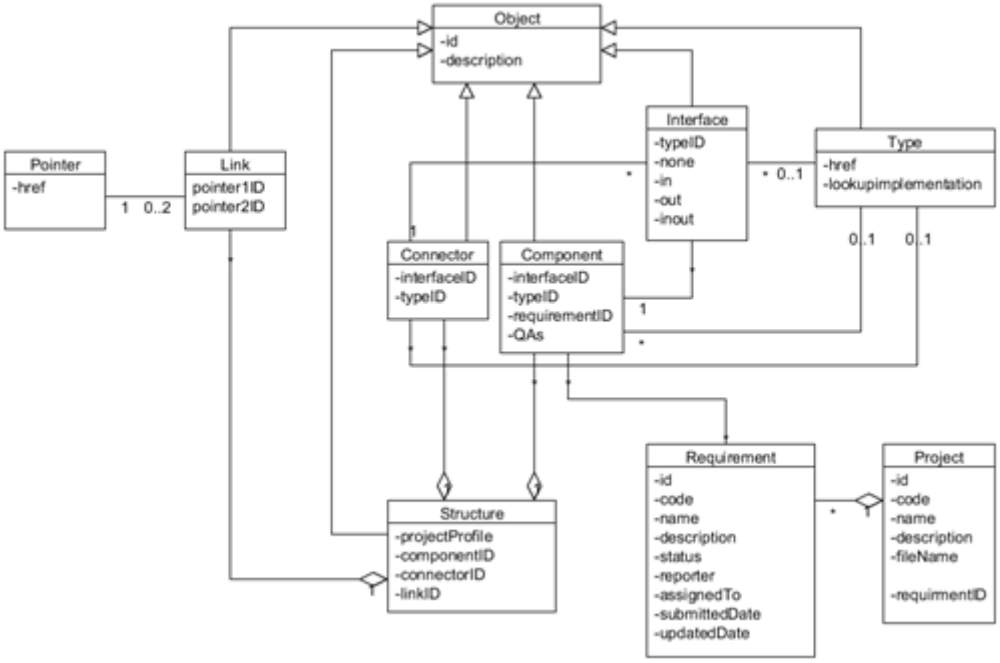
ชื่อยูสเคส	ยูสเคสตรวจจับความเปลี่ยนแปลงและผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์
ผู้กระทำ	สถาปนิกซอฟต์แวร์,นักวิเคราะห์ระบบ นักออกแบบสถาปัตยกรรมซอฟต์แวร์
รายละเอียดยูสเคส	การตรวจจับความเปลี่ยนแปลงและสรุปผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ ผลลัพธ์ของการตรวจจับ สามารถแสดงเป็นรายงาน แบ่งออกเป็นสามส่วน คือ 1. รายงานผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ ซึ่งรวบรวมข้อมูลรายละเอียดข้อกำหนดความต้องการซอฟต์แวร์ คุณลักษณะเชิงคุณภาพ หรือชุดคำสั่งใดๆ ที่ได้รับผลกระทบจากการเปลี่ยนแปลง 2. รายงานสรุปผลความเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ 3. รายงานแสดงรายละเอียดการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์
เงื่อนไขก่อนหน้า	แฟ้มข้อมูลนำเข้าที่ใช้ตรวจจับต้องมีรูปแบบเป็นภาษาเอ็กซ์เอดีแอลเวอร์ชันสอง
ขั้นตอน	1. ผู้ใช้งานเข้าสู่หน้าจอการตรวจจับความเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ 2. ผู้ใช้งานเลือกแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์ที่ต้องการ 3. ผู้ใช้งานเลือกยืนยันการตรวจจับ

ตารางที่ 4.7 คำอธิบายยูสเคสนำเข้าสถาปัตยกรรมซอฟต์แวร์ต้นฉบับและสถาปัตยกรรมซอฟต์แวร์ที่ถูกแก้ไข

ชื่อยูสเคส	นำเข้าสถาปัตยกรรมซอฟต์แวร์ต้นฉบับและสถาปัตยกรรมซอฟต์แวร์ที่ถูกแก้ไข
ผู้กระทำ	สถาปนิกซอฟต์แวร์,นักวิเคราะห์ระบบ นักออกแบบสถาปัตยกรรมซอฟต์แวร์
รายละเอียดยูสเคส	การนำเข้าสถาปัตยกรรมซอฟต์แวร์ต้นฉบับและสถาปัตยกรรมซอฟต์แวร์ที่ถูกแก้ไข เพื่อใช้ในการตรวจจับการเปลี่ยนแปลงและผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ ซึ่งรวบรวมข้อมูลรายละเอียดข้อกำหนดความต้องการซอฟต์แวร์ คุณลักษณะเชิงคุณภาพ หรือชุดคำสั่งใดๆ ที่ได้รับผลกระทบจากการเปลี่ยนแปลงที่ตรวจจับได้แสดงผลในรูปแบบรายงาน
เงื่อนไขก่อนหน้า	แฟ้มข้อมูลนำเข้าสถาปัตยกรรมซอฟต์แวร์ต้นฉบับและสถาปัตยกรรมซอฟต์แวร์ที่ถูกแก้ไขที่ใช้ตรวจจับต้องมีรูปแบบเป็นภาษาเอ็กซ์เอดีแอลเวอร์ชันสอง และมีการกำหนดคำอธิบายในการออกแบบส่วนประกอบ
ขั้นตอน	1. ผู้ใช้งานเข้าสู่หน้าจอการตรวจจับความเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ 2. ผู้ใช้งานเลือกแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์ต้นฉบับและสถาปัตยกรรมซอฟต์แวร์ที่ถูกแก้ไขที่ต้องการ 3. ผู้ใช้งานเลือกยืนยันการนำเข้า
เงื่อนไขภายหลัง	กรณีที่เครื่องมือไม่พบโครงสร้างสถาปัตยกรรมซอฟต์แวร์ หรือแฟ้มข้อมูลนำเข้าไม่เป็นไปตามรูปแบบที่กำหนดเครื่องมือจะแสดงข้อความแจ้งเตือนให้ผู้ใช้งานทราบ
เงื่อนไขภายหลัง	กรณีที่เครื่องมือไม่พบโครงสร้างสถาปัตยกรรมซอฟต์แวร์ หรือแฟ้มข้อมูลนำเข้าไม่เป็นไปตามรูปแบบที่กำหนดเครื่องมือจะแสดงข้อความแจ้งเตือนให้ผู้ใช้งานทราบ

4.2.2 แผนภาพคลาส

เครื่องมือสำหรับการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์มีองค์ประกอบและความสัมพันธ์ของข้อมูลที่เกี่ยวข้องกัน โดยสามารถแสดงเป็นแผนภาพคลาสได้ดังรูป ที่ 4.3



รูปที่ 4.3 แผนภาพคลาสของเครื่องมือที่ใช้ในการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์

แผนภาพคลาสในรูปที่ 4.3 มีรายละเอียดดังต่อไปนี้

- คลาส Object คือคลาส ที่ใช้จัดเก็บข้อมูลเบื้องต้นของจาวาออบเจ็กต์ที่ใช้ในระบบ
- คลาส Pointer คือคลาส ที่ใช้จัดเก็บข้อมูลของพอยเตอร์ที่ใช้กับเส้นเชื่อม
- คลาส Link คือคลาส ที่ใช้จัดเก็บข้อมูลเส้นเชื่อมของโครงสร้างสถาปัตยกรรม
- คลาส Connector คือคลาส ที่ใช้จัดเก็บข้อมูลตัวเชื่อมของโครงสร้างสถาปัตยกรรม
- คลาส Component คือคลาส ที่ใช้จัดเก็บข้อมูลส่วนประกอบของโครงสร้างสถาปัตยกรรม
- คลาส Interface คือคลาส ที่ใช้จัดเก็บข้อมูลจุดเชื่อมของโครงสร้างสถาปัตยกรรม
- คลาส Type คือคลาส ที่ใช้จัดเก็บข้อมูลประเภทที่ใช้กับส่วนประกอบ ตัวเชื่อม หรือจุดเชื่อมของโครงสร้างสถาปัตยกรรม
- คลาส Structure คือคลาส ที่ใช้จัดเก็บข้อมูลเบื้องต้นของโครงสร้างสถาปัตยกรรม
- คลาส Requirement คือคลาส ที่ใช้จัดเก็บข้อมูลข้อกำหนดความต้องการที่เกี่ยวข้องกับส่วนประกอบโครงสร้างสถาปัตยกรรม
- คลาส Project คือคลาส ที่ใช้จัดเก็บข้อมูลเบื้องต้นของโครงการซอฟต์แวร์

ตารางที่ 4.8 รายละเอียดตารางความสัมพันธ์ระหว่างคุณลักษณะกับแผนภาพคลาสจาวาออบเจ็กต์

ชื่อคลาส	รายละเอียด
Object	เก็บข้อมูลจาวาออบเจ็กต์ที่ใช้กับเครื่องมือ
Pointer	เก็บข้อมูลพอยเตอร์ที่ใช้กับเส้นเชื่อม
Link	เก็บข้อมูลเส้นเชื่อมสถาปัตยกรรม
Connector	เก็บข้อมูลตัวเชื่อมสถาปัตยกรรม
Component	เก็บข้อมูลส่วนประกอบสถาปัตยกรรม
Interface	เก็บข้อมูลจุดเชื่อมสถาปัตยกรรม
Type	เก็บข้อมูลประเภทส่วนประกอบ, ประเภทตัวเชื่อม หรือประเภทจุดเชื่อมสถาปัตยกรรม
Structure	เก็บข้อมูลโครงสร้างสถาปัตยกรรม
Requirement	เก็บข้อมูลข้อกำหนดความต้องการที่เกี่ยวข้องกับส่วนประกอบสถาปัตยกรรม
Project	เก็บข้อมูลข้อมูลเบื้องต้นของโครงการซอฟต์แวร์

รายละเอียดข้อมูลในแต่ละคลาสแสดงไว้ดังตารางที่ 4.9 – 4.18 ดังต่อไปนี้

ตารางที่ 4.9 โครงสร้างข้อมูลของคลาส Object

คุณลักษณะ	รายละเอียด	ชนิด	ค่าคีย์
id	ค่าคีย์ของคลาส Object	String	Yes
description	คำอธิบายออบเจ็กต์	String	

ตารางที่ 4.10 โครงสร้างข้อมูลของคลาส Pointer

คุณลักษณะ	รายละเอียด	ชนิด	ค่าคีย์
href	ค่าไฮเปอร์ลิงค์ ที่ถูกนำไปใช้งานกับเส้นเชื่อมของโครงสร้างสถาปัตยกรรม	String	

ตารางที่ 4.11 โครงสร้างข้อมูลของคลาส Link

คุณลักษณะ	รายละเอียด	ชนิด	ค่าคีย์
pointer1ID	ค่าคีย์ของคลาส Pointer จุดที่หนึ่ง	String	
pointer2ID	ค่าคีย์ของคลาส Pointer จุดที่สอง	String	

ตารางที่ 4.12 โครงสร้างข้อมูลของคลาส Connector

คุณลักษณะ	รายละเอียด	ชนิด	ค่าคีย์
interfaceID	ค่าคีย์ของคลาส Interface	List	
typeID	ค่าคีย์ของคลาส Type	List	

ตารางที่ 4.13 โครงสร้างข้อมูลของคลาส Component

คุณลักษณะ	รายละเอียด	ชนิด	ค่าคีย์
interfaceID	ค่าคีย์ของคลาส Interface	List	
typeID	ค่าคีย์ของคลาส Type	List	
requirementID	ค่าคีย์ของคลาส Requirement	List	
QAs	ค่านิพจน์ที่ใช้แสดงคุณลักษณะเชิงคุณภาพ	List <Boolean>	

ตารางที่ 4.14 โครงสร้างข้อมูลของคลาส Interface

คุณลักษณะ	รายละเอียด	ชนิด	ค่าคีย์
typeID	ค่าคีย์ของคลาส Type	List	
none	จำนวนจุดเชื่อมต่อที่ไม่ได้ระบุทิศทางการเชื่อมต่อ	Integer	
in	จำนวนจุดเชื่อมต่อที่ระบุทิศทางการเชื่อมต่อเข้า	Integer	
out	จำนวนจุดเชื่อมต่อที่ระบุทิศทางการเชื่อมต่อออก	Integer	
inout	จำนวนจุดเชื่อมต่อที่ระบุทิศทางการเชื่อมต่อเข้าและออก	Integer	

ตารางที่ 4.15 โครงสร้างข้อมูลของคลาส Type

คุณลักษณะ	รายละเอียด	ชนิด	ค่าคีย์
href	ค่าไฮเปอร์ลิงค์ ที่ถูกนำไปใช้งานกับส่วนประกอบ, ตัวเชื่อม หรือจุดเชื่อมของโครงสร้างสถาปัตยกรรม	String	
lookupimplementation	ค่าคลาสแพ็กเกจที่มีการนำไปใช้งาน	String	

ตารางที่ 4.16 โครงสร้างข้อมูลของคลาส Structure

คุณลักษณะ	รายละเอียด	ชนิด	ค่าคีย์
projectProfile	ชื่อโครงการที่สถาปัตยกรรมมีความเกี่ยวข้อง	String	
componentID	ค่าคีย์ของคลาส Component	List	
connectorID	ค่าคีย์ของคลาส Connector	List	
linkID	ค่าคีย์ของคลาส Link	List	

ตารางที่ 4.17 โครงสร้างข้อมูลของคลาส Requirement

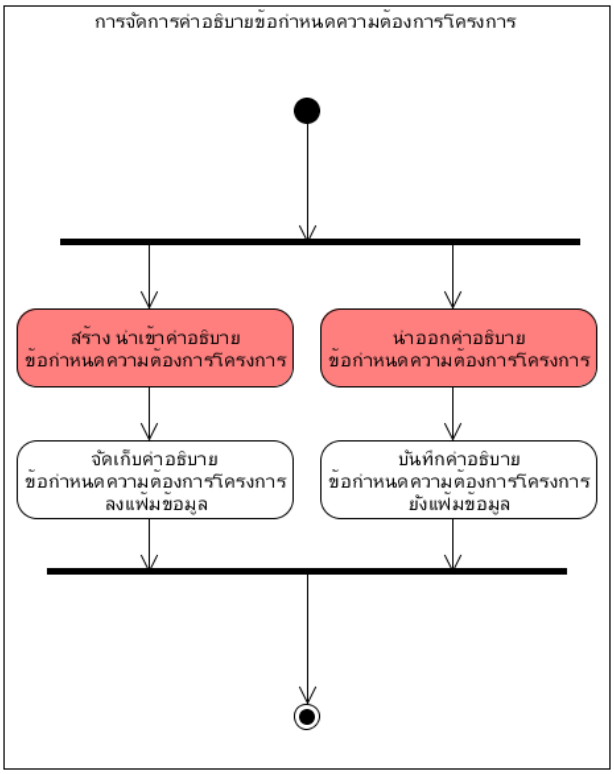
คุณลักษณะ	รายละเอียด	ชนิด	ค่าคีย์
id	ค่าคีย์ของคลาส Requirement	String	Yes
code	รหัสความต้องการ	String	
name	ชื่อความต้องการ	String	
description	คำอธิบายความต้องการ	String	
status	สถานะความต้องการ	String	
reporter	ชื่อผู้รายงานความต้องการ	String	
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	
submittedDate	วันที่สร้าง	Date	
updatedDate	วันที่ปรับปรุงข้อมูล	Date	

ตารางที่ 4.18 โครงสร้างข้อมูลของคลาส Project

คุณลักษณะ	รายละเอียด	ชนิด	ค่าคีย์
id	ค่าคีย์ของคลาส Project	String	Yes
code	รหัสโครงการ	String	
name	ชื่อโครงการ	String	
description	คำอธิบายโครงการ	String	
fileName	ชื่อแฟ้มข้อมูล	String	
requirementID	ค่าคีย์ของคลาส Requirement	List	

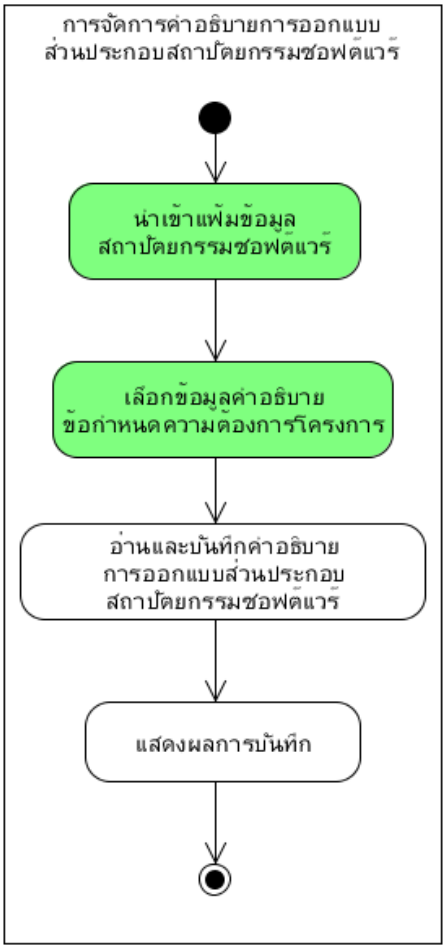
4.2.3 แผนภาพกิจกรรม

เครื่องมือสำหรับการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ มีกิจกรรม 3 ส่วน คือ กิจกรรมการจัดการคำอธิบายข้อกำหนดความต้องการโครงการ กิจกรรมการจัดการคำอธิบายการออกแบบ ส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ และกิจกรรมการตรวจจับความเปลี่ยนแปลงและสรุปผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ ตามแผนภาพกิจกรรม รายละเอียดดังนี้



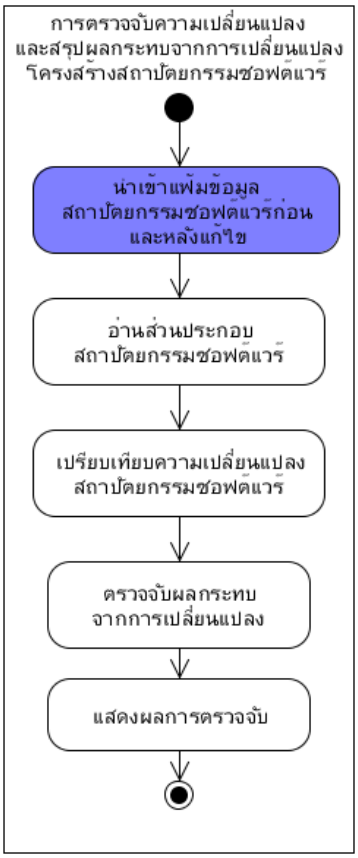
รูปที่ 4.4 แผนภาพกิจกรรมการจัดการคำอธิบายข้อกำหนดความต้องการโครงการ

1) **การจัดการคำอธิบายข้อกำหนดความต้องการโครงการ** (Project Requirements Description Management) มีจุดประสงค์เพื่อให้ผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์ สามารถจัดการข้อมูลคำอธิบายข้อกำหนดโครงการเบื้องต้น ยกตัวอย่างเช่น ชื่อโครงการ, คำอธิบายโครงการ รวมถึงรายการข้อกำหนดความต้องการที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์ ข้อมูลที่ได้รับการบันทึกจะถูกจัดเก็บในรูปแบบแฟ้มข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการที่เป็นแฟ้มข้อมูลประเภทซีเอสวี เพื่อนำไปใช้ในขั้นตอนการระบุคำอธิบายการออกแบบต่อไป ทั้งนี้ ผู้ใช้งานยังสามารถนำเข้า (Import) และส่งออก (Export) ข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการที่ได้ทำการบันทึกไว้ เพื่อนำข้อมูลไปใช้งานในขั้นตอนอื่นๆของกระบวนการพัฒนาซอฟต์แวร์ต่อไปได้อีกด้วย



รูปที่ 4.5 แผนภาพกิจกรรมการจัดการคำอธิบายการออกแบบ
ส่วนประกอบสถาปัตยกรรมซอฟต์แวร์

2) **การจัดการคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ (Design Description Management)** มีจุดประสงค์เพื่อให้สถาปนิกซอฟต์แวร์ หรือนักออกแบบสถาปัตยกรรมซอฟต์แวร์ สามารถจัดการคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ ซึ่งใช้เป็นข้อมูลเบื้องต้นในการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ ทั้งนี้ ข้อมูลที่ผู้ใช้งานสามารถกำหนดให้กับส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ได้ แบ่งออกเป็นสองส่วน คือ รายการข้อกำหนดความต้องการที่เกี่ยวข้องกับการออกแบบส่วนประกอบ (ผู้ใช้งานเลือกข้อมูลจากส่วนของการจัดการคำอธิบายข้อกำหนดความต้องการโครงการในข้อที่หนึ่ง และคุณลักษณะเชิงคุณภาพที่เกี่ยวข้องกับการออกแบบส่วนประกอบ โดยมีทั้งหมด 6 ประเภท คือ หน้าที่การทำงาน (Functionality), ความน่าเชื่อถือ (Reliability), การใช้งาน (Usability), ประสิทธิภาพ (Efficiency), การบำรุงรักษา (Maintainability) และความสะดวกในการพกพา (Portability)



รูปที่ 4.6 แผนภาพกิจกรรมการตรวจจับความเปลี่ยนแปลงและสรุปผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์

3) **การตรวจจับความเปลี่ยนแปลงและสรุปผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์** (Software Architecture Changes and Effect Detection) ขั้นตอนนี้เป็นการตรวจจับความเปลี่ยนแปลงและสรุปผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ เริ่มต้นจากผู้ใช้ทำการนำเข้าแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์ก่อนและหลังการแก้ไข หลังจากนั้น เครื่องมือจะทำการอ่านโครงสร้างข้อมูลสถาปัตยกรรมซอฟต์แวร์ในรูปแบบภาษาเอ็กซ์เอดีแอลเพื่อจัดเก็บข้อมูลส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ที่ใช้ในการตรวจจับผลกระทบไปยังจาวาออบเจ็กต์ที่ออกแบบไว้ จากนั้นจึงผ่านขั้นตอนการตรวจจับความเปลี่ยนแปลงและผลกระทบจากการเปลี่ยนแปลงที่ได้กำหนดไว้ก่อนหน้า

4.3 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ

สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ มีรายละเอียดแสดงได้ดังต่อไปนี้

1) **ฮาร์ดแวร์** (Hardware)

- เครื่องคอมพิวเตอร์แบบโน้ตบุ๊ก (Notebook) หน่วยประมวลผลอินเทลคอร์ไอไฟว์ความเร็วสัญญาณนาฬิกา 2.30 กิกะเฮิร์ท (Intel CORE i5 2.30 GHz)
- หน่วยความจำสำรอง (RAM) ขนาด 4 กิกะไบต์ (4 GB)
- ฮาร์ดดิสก์ (Hard disk) ขนาด 500 กิกะไบต์ (500 GB)

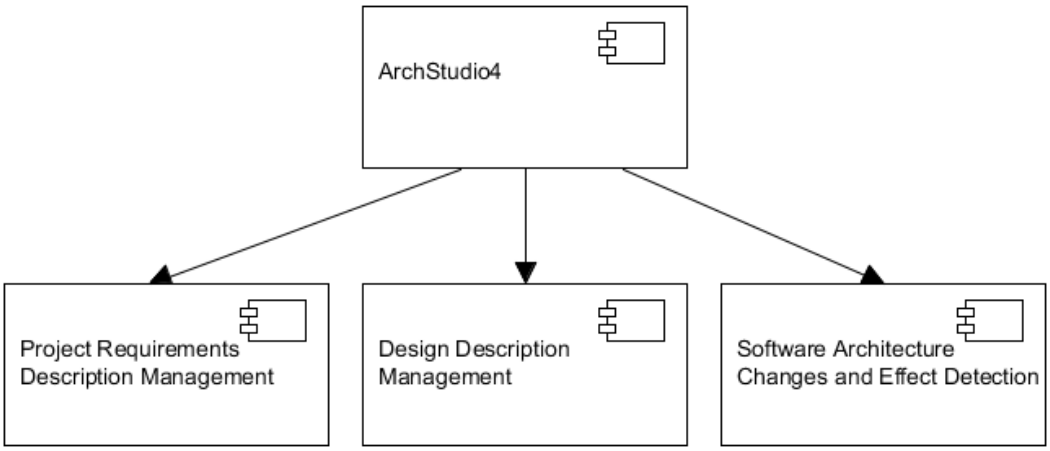
2) **ซอฟต์แวร์** (Software)

- ระบบปฏิบัติการ (Operation System) ไมโครซอฟต์วินโดวส์ 7 (Microsoft Windows 7)
- โปรแกรมอาร์คสตูดิโอ เวอร์ชัน 4 (ArchStudio 4)
- โปรแกรมอีคลิปส์ อินดีโก ไอดีอี (Eclipse Indigo IDE)
- โปรแกรมอีดิทพลัส เวอร์ชัน 3.11 (EditPlus 3.11)
- โปรแกรมแจสเปอร์ รีพอร์ต เวอร์ชัน 4.8.0 (JasperReport 4.8.0)

4.4 โครงสร้างส่วนต่อประสานกับผู้ใช้เครื่องมือ

โครงสร้างส่วนต่อประสานกับผู้ใช้เครื่องมือในการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ อธิบายด้วยแผนภาพส่วนประกอบ (Component Diagram) ซึ่งเป็นแผนภาพที่ใช้ในการแสดงความสัมพันธ์ระหว่างส่วนประกอบต่างๆ ของเครื่องมือที่ได้ทำการพัฒนาขึ้นจากการเพิ่มเติมส่วนต่อขยายของเครื่องมือ ArchStudio 4 โดยสามารถแบ่งการทำงานหลักออกได้ทั้งหมดสามส่วน คือ ส่วนการจัดการคำอธิบายข้อกำหนดความต้องการโครงการ, ส่วน

การจัดการคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ และส่วนการตรวจจับความเปลี่ยนแปลงและสรุปผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ โดยแผนภาพส่วนประกอบแสดงส่วนต่อประสานกับผู้ใช้งานเครื่องมือ สามารถแสดงได้ดังรูปที่ 4.7



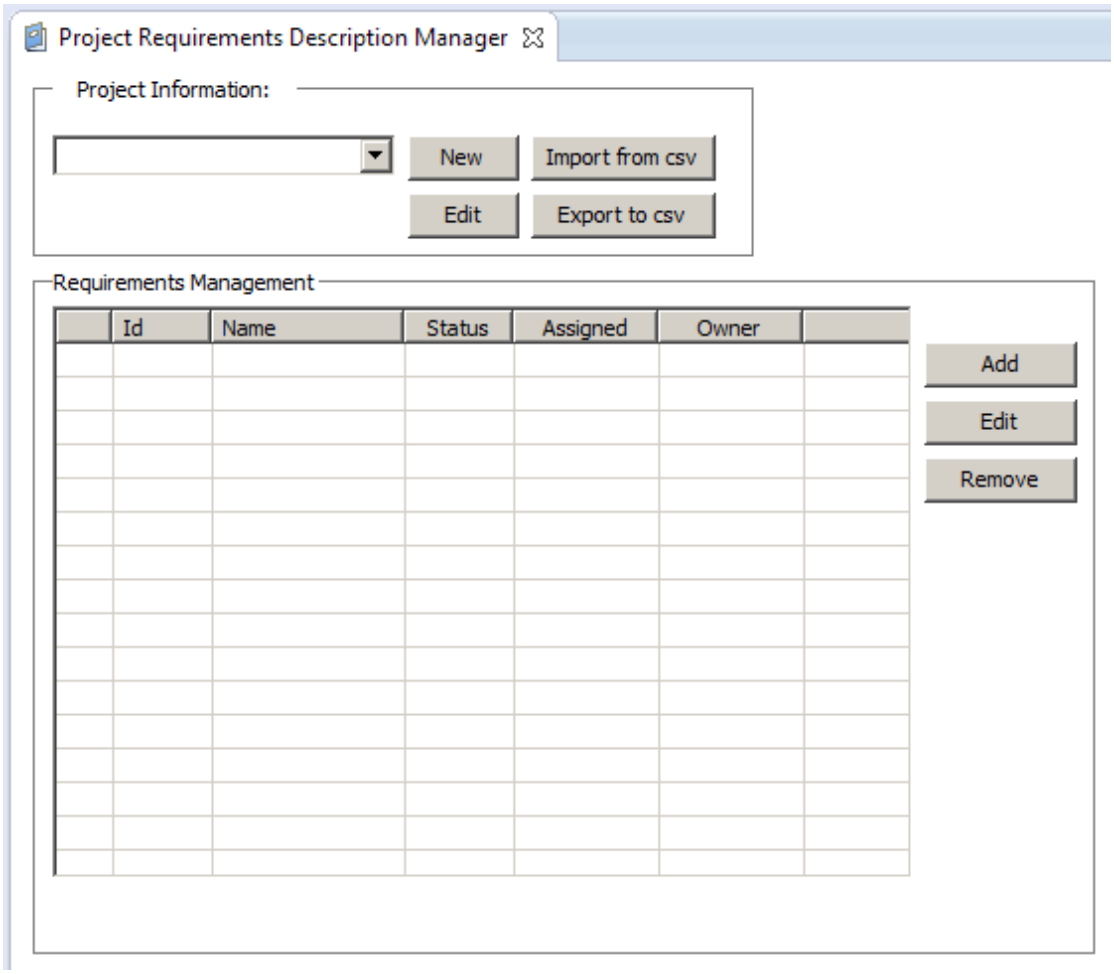
รูปที่ 4.7 แผนภาพส่วนประกอบของเครื่องมือ

จากรูปที่ 4.7 ส่วนประกอบที่แสดงในรูปสามารถแทนได้ด้วยหน้าที่การทำงานของเครื่องมือ โดยหน้าจอการทำงานจะประกอบไปด้วย

1. ส่วนการจัดการคำอธิบายข้อกำหนดความต้องการโครงการ
2. ส่วนการจัดการคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์
3. ส่วนการตรวจจับความเปลี่ยนแปลงและสรุปผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์

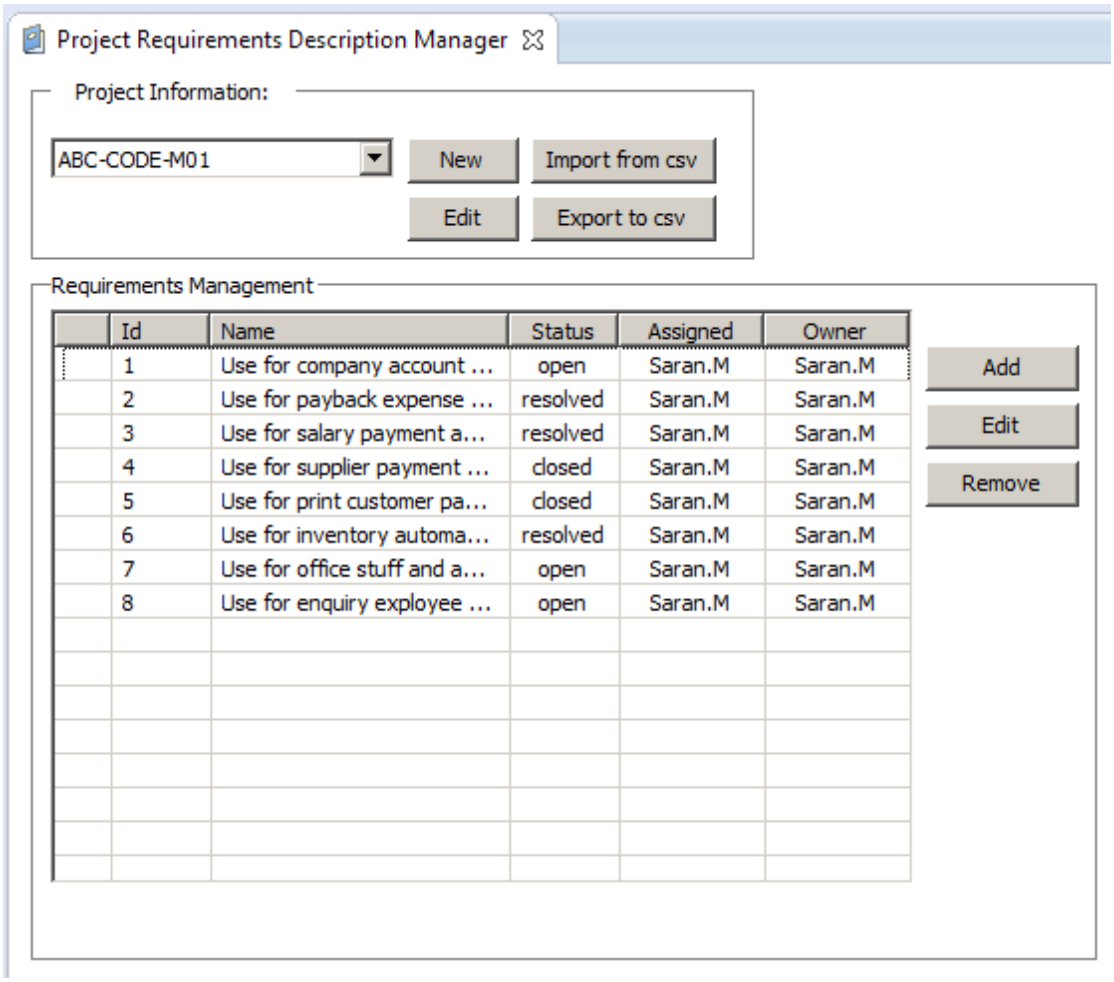
4.4.1 ส่วนการจัดการคำอธิบายข้อกำหนดความต้องการโครงการ

ผู้ใช้งานสามารถสร้าง แก้ไข นำเข้าและนำออก ข้อมูลเบื้องต้นของโครงการซอฟต์แวร์ได้ นอกจากนี้ ผู้ใช้งานยังสามารถเพิ่ม แก้ไข ลบ รายการข้อกำหนดความต้องการที่เกี่ยวข้องกันในแต่ละโครงการซอฟต์แวร์ จากหน้าจอที่ใช้ในการจัดการข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ ดังแสดงในรูปที่ 4.8 - 4.9



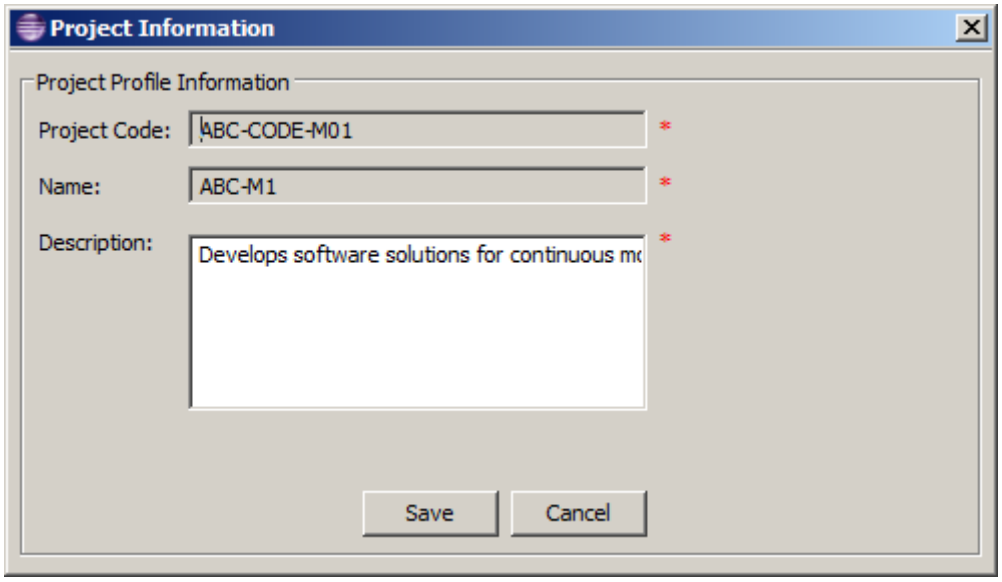
รูปที่ 4.8 ตัวอย่างส่วนต่อประสานหลักกับผู้ใช้งานสำหรับการจัดการข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ

หลังจากผู้ใช้งานเข้ามายังส่วนการทำงานของการจัดการข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการ เครื่องมือจะทำการแสดงหน้าจอสำหรับการจัดการข้อมูลต่างๆ ซึ่งแบ่งแยกการทำงานออกเป็นสองส่วน คือ ส่วนของการจัดการข้อมูลเบื้องต้นของโครงการซอฟต์แวร์ และส่วนการจัดการรายการข้อกำหนดความต้องการที่เกี่ยวข้องกับโครงการซอฟต์แวร์ โดยมีข้อมูลรายการข้อกำหนดความต้องการที่นำมาแสดงผลในหน้าจอหลักแยกตามแต่ละโครงการซอฟต์แวร์ ได้แก่ รหัสอ้างอิงความต้องการ, ชื่อความต้องการ, สถานะ, ผู้ได้รับมอบหมาย และผู้รายงานความต้องการ



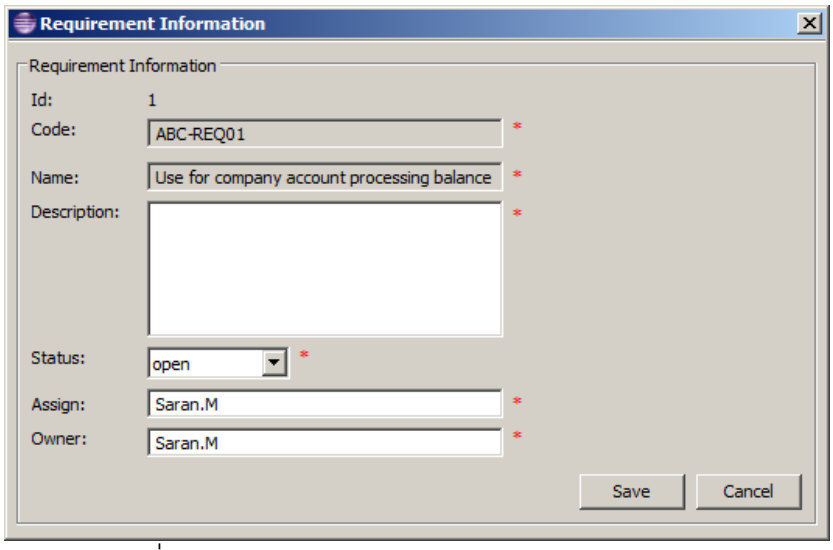
รูปที่ 4.9 ตัวอย่างส่วนต่อประสานกับผู้ใช้งานแสดงผลรายการ
ข้อกำหนดความต้องการที่เกี่ยวกับโครงการซอฟต์แวร์

หลังจากผู้ใช้งานเลือกข้อมูลโครงการผ่านการกำหนดรหัสโครงการ เพื่อทำการจัดการข้อมูลแล้ว เครื่องมือจะนำข้อมูลที่เกี่ยวกับโครงการมาแสดงในหน้าจอ โดยข้อมูลทั้งหมดสามารถนำมาใช้ในการระบุข้อกำหนดความต้องการโครงการซอฟต์แวร์ ซึ่งมีข้อมูลให้ผู้ใช้งานสามารถบันทึก แก้ไขข้อมูล ได้แก่ รหัสโครงการ, ชื่อโครงการ และคำอธิบายโครงการ รวมไปถึงรายการข้อกำหนดความต้องการที่เกี่ยวข้องกับโครงการซอฟต์แวร์ รายละเอียดสามารถแสดงได้ดังรูปที่ 4.10 – 4.11



รูปที่ 4.10 ตัวอย่างส่วนต่อประสานกับผู้ใช้งานสำหรับการบันทึกข้อมูลโครงการเบื้องต้น

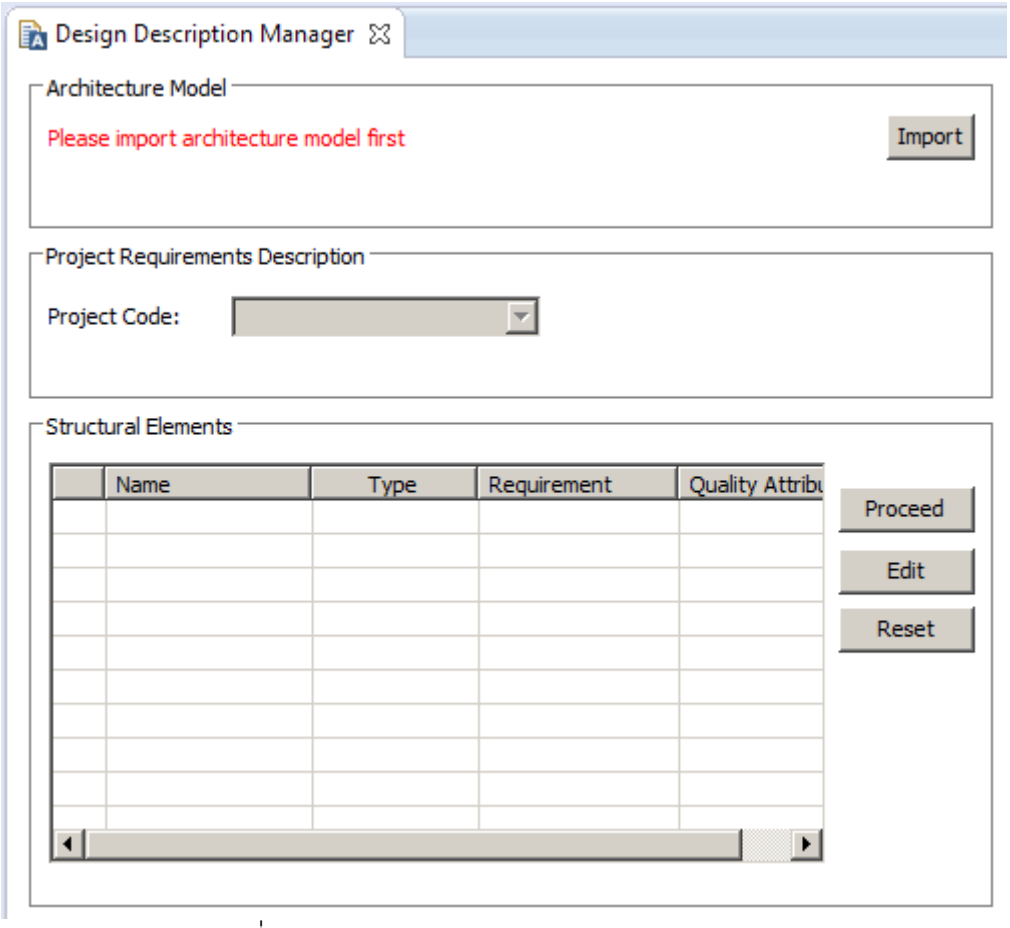
เมื่อผู้ใช้งานได้ทำการบันทึกข้อมูลโครงการเบื้องต้นเรียบร้อยแล้ว ผู้ใช้งานยังสามารถทำการเพิ่ม แก้ไข ลบ รายการข้อกำหนดความต้องการที่เกี่ยวข้องกับโครงการซอฟต์แวร์ โดยรายละเอียดของรายการข้อกำหนดความต้องการที่ผู้ใช้งานสามารถบันทึกได้ คือ รหัสความต้องการ, ชื่อความต้องการ, คำอธิบาย, สถานะ, ผู้ที่ได้รับมอบหมาย และผู้กำหนดความต้องการ รายละเอียดดังกล่าวสามารถแสดงได้ดังรูปที่ 4.9



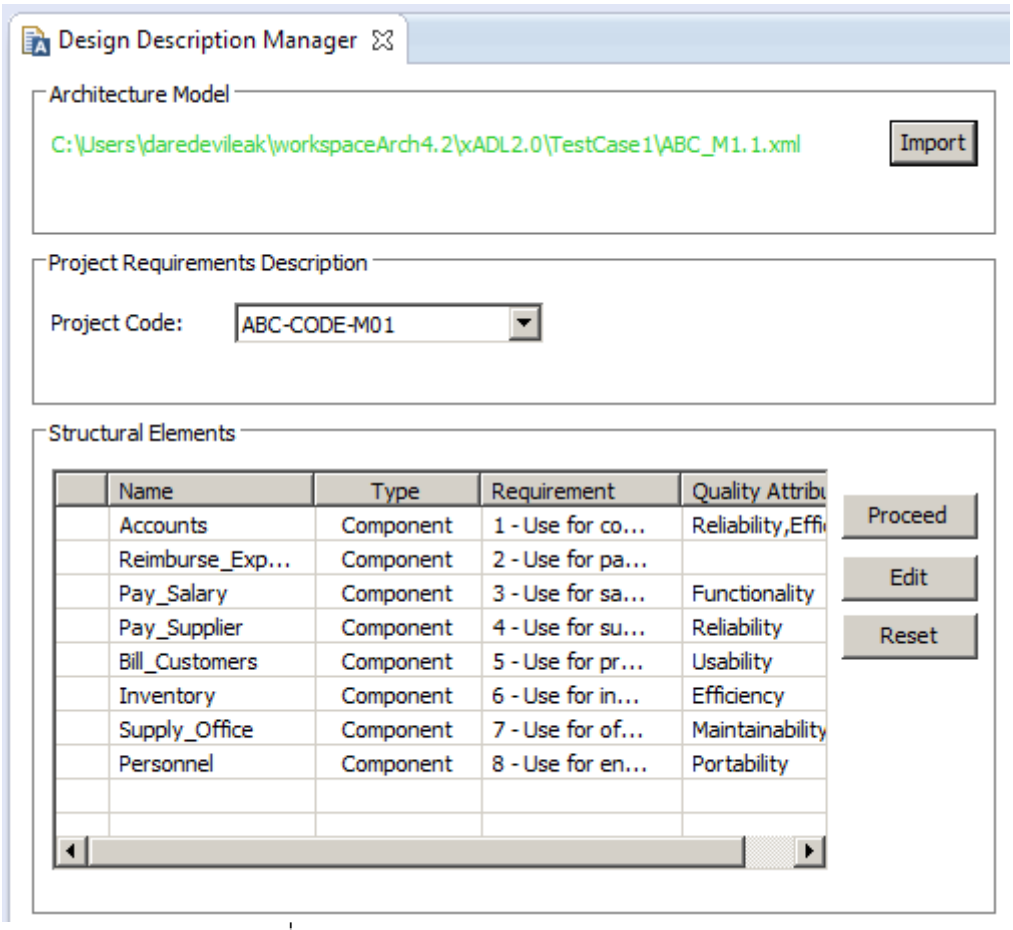
รูปที่ 4.11 ตัวอย่างส่วนต่อประสานกับผู้ใช้งานสำหรับ
การบันทึกข้อมูลข้อกำหนดความต้องการของโครงการ

4.4.2 ส่วนการจัดการคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์

ผู้ใช้งานสามารถจัดการคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ผ่านหน้าจอ โดยผู้ใช้งานจะต้องทำการเลือกแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์ที่ได้ทำการออกแบบไว้ หลังจากนั้นเครื่องมือจะแสดงผลส่วนประกอบที่ได้รับการออกแบบ และคำอธิบายการออกแบบส่วนประกอบเดิมที่ได้รับการกำหนดไว้ เพื่อให้ผู้ใช้งานเลือกสำหรับการจัดการข้อมูล ผู้ใช้งานยังสามารถเลือกข้อมูลคำอธิบายข้อกำหนดความต้องการโครงการที่ใช้สำหรับการออกแบบสถาปัตยกรรมซอฟต์แวร์ เพื่อนำรายการข้อกำหนดความต้องการที่เกี่ยวข้องกับสถาปัตยกรรม นำใช้มากำหนดเป็นคำอธิบายในการออกแบบส่วนประกอบต่างๆ เพิ่มเติม รายละเอียดดังกล่าวสามารถแสดงได้ดังรูปที่ 4.12 - 4.14

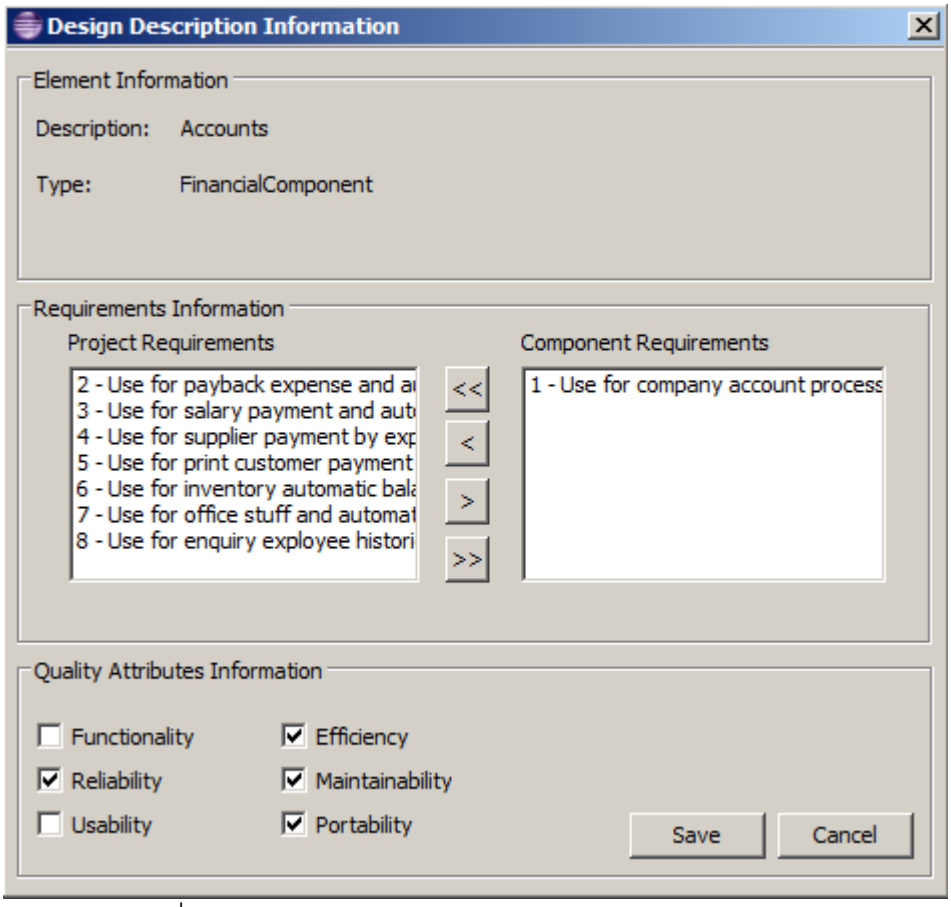


รูปที่ 4.12 ตัวอย่างส่วนต่อประสานหลักกับผู้ใช้งาน
สำหรับจัดการคำอธิบายการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์



รูปที่ 4.14 ตัวอย่างส่วนต่อประสานกับผู้ใช้งาน
แสดงส่วนประกอบสถาปัตยกรรมซอฟต์แวร์หลังจากนำเข้าแฟ้มข้อมูล

จากตัวอย่างส่วนต่อประสานกับผู้ใช้งานดังรูปที่ 4.7 เครื่องมือแสดงรายละเอียดส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ รหัสโครงการ ABC-CODE-M01 ซึ่งประกอบไปด้วยส่วนประกอบทั้งหมด 8 ส่วนประกอบ ได้แก่ 1. Accounts 2. Reimburse_Expesnse 3. Pay_Salary 4. Pay_Supplier 5. Bill_Customers 6. Inventory 7. Supply_Office และ 8. Personnel อีกทั้งยังแสดงคำอธิบายการออกแบบส่วนประกอบที่มีรายการข้อกำหนดความต้องการ และคุณลักษณะเชิงคุณภาพ ที่ใช้ในการออกส่วนประกอบแต่ละส่วนประกอบ ยกตัวอย่างเช่น ส่วนประกอบ Accounts มีรายการข้อกำหนดความต้องการ คือ รายการข้อกำหนดความต้องการหมายเลข 1 - Use for company account processing balance และคุณลักษณะเชิงคุณภาพด้านความน่าเชื่อถือ ด้านประสิทธิภาพ การบำรุงรักษาและความสะดวกในการพกพา ซึ่งเกี่ยวข้องกับเหตุผลในการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ เป็นต้น

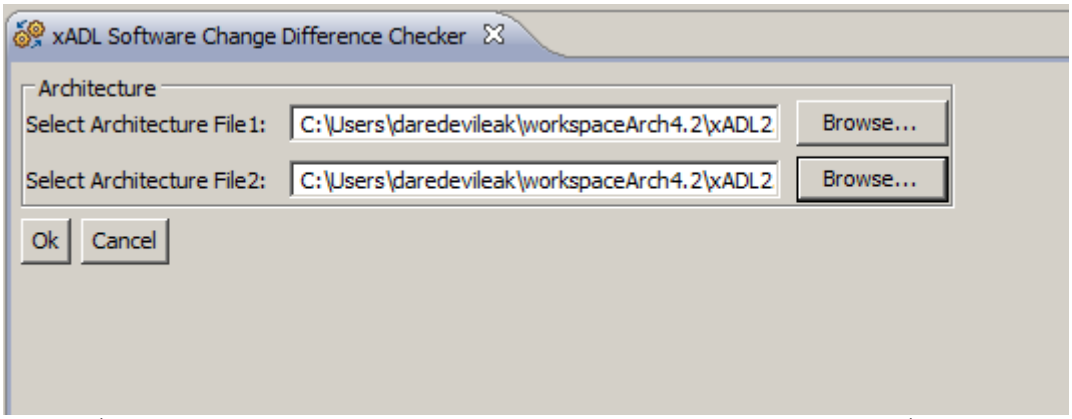


รูปที่ 4.14 ตัวอย่างส่วนต่อประสานกับผู้ใช้งานสำหรับการ
กำหนดคำอธิบายการออกแบบให้กับส่วนประกอบ

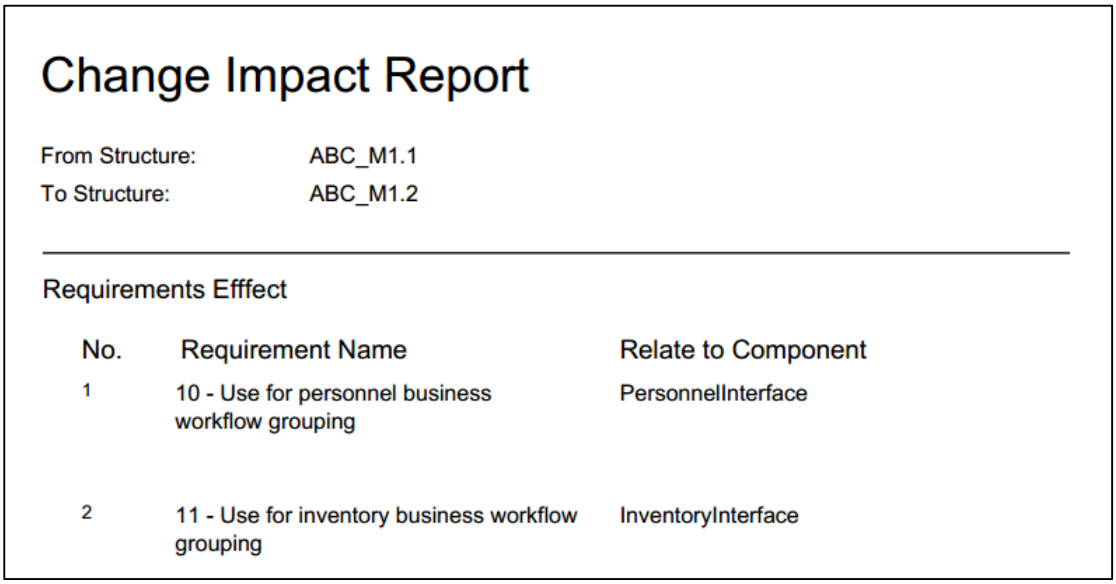
จากตัวอย่างส่วนต่อประสานกับผู้ใช้งานดังรูปที่ 4.12 เครื่องมือแสดงรายละเอียดรายการข้อกำหนดความต้องการและคุณลักษณะเชิงคุณภาพ ที่ใช้เป็นคำอธิบายการออกแบบส่วนประกอบ Accounts คือ รายการข้อกำหนดความต้องการหมายเลข 1 - Use for company account processing balance และคุณลักษณะเชิงคุณภาพด้านความน่าเชื่อถือ ด้านประสิทธิภาพ การบำรุงรักษา และความสะดวกในการพกพา ซึ่งเกี่ยวข้องกับเหตุผลในการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์

4.4.3 ส่วนการตรวจจับความเปลี่ยนแปลงและสรุปผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์

ผู้ใช้งานสามารถทำการตรวจจับความเปลี่ยนแปลงและตรวจสอบผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ ผ่านหน้าจอสำหรับการตรวจจับความเปลี่ยนแปลงและสรุปผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ โดยผู้ใช้งานจะต้องทำการเลือกแฟ้มข้อมูลสถาปัตยกรรมซอฟต์แวร์ที่ได้ออกแบบไว้ ในรูปแบบภาษาเอ็กซ์เอดีแอล ส่วนแรกเป็นเวอร์ชันตั้งต้น และเลือกแฟ้มข้อมูลในรูปแบบภาษาเอ็กซ์เอดีแอลส่วนที่ได้มีการทำการปรับปรุง แก้ไข จากส่วนแรก จากนั้นเลือกยืนยันเพื่อทำการตรวจจับ หลังจากเครื่องมือทำงานเสร็จสิ้น เครื่องมือจะทำการแสดงผลลัพธ์การตรวจจับความเปลี่ยนแปลง และสรุปผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ในรูปแบบรายงาน ซึ่งแบ่งออกเป็นสามส่วน ได้แก่ รายงานสรุปส่วนประกอบที่ได้รับผลกระทบจากการเปลี่ยนแปลง รายงานสรุปผลความเปลี่ยนแปลง และรายงานแสดงรายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ โดยรายละเอียดดังกล่าวสามารถแสดงได้ดังรูปที่ 4.15 - 4.18



รูปที่ 4.15 ตัวอย่างส่วนต่อประสานกับผู้ใช้งานสำหรับการตรวจจับความเปลี่ยนแปลงและสรุปผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์



Change Impact Report

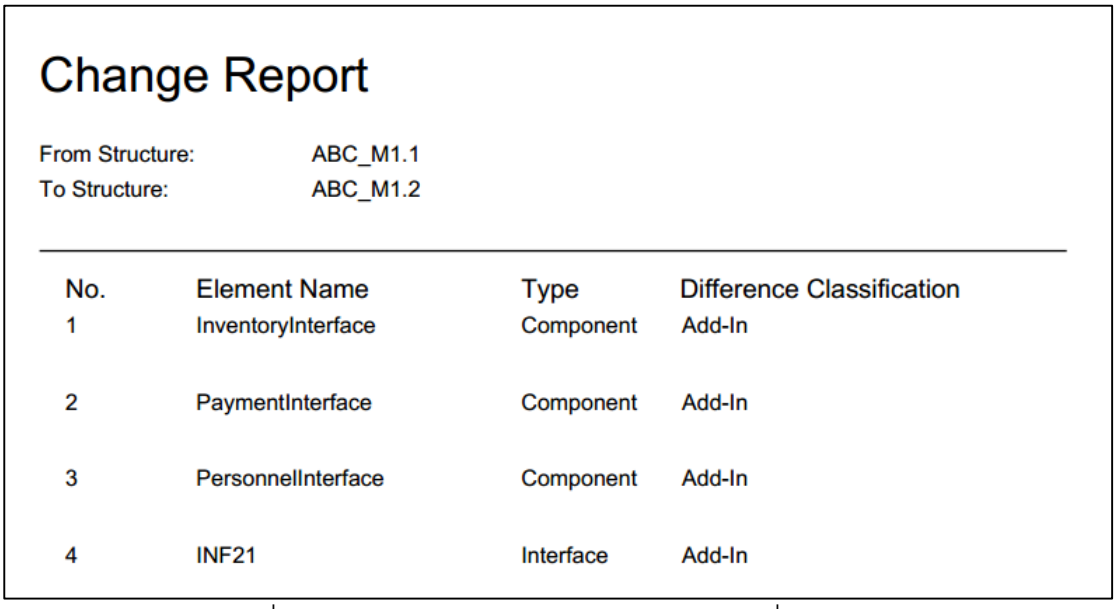
From Structure: ABC_M1.1
To Structure: ABC_M1.2

Requirements Efffect

No.	Requirement Name	Relate to Component
1	10 - Use for personnel business workflow grouping	PersonnelInterface
2	11 - Use for inventory business workflow grouping	InventoryInterface

รูปที่ 4.16 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ในรูปแบบรายงาน

ตัวอย่างรายงานแสดงผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ จากสถาปัตยกรรมซอฟต์แวร์ ABC_M1.1 เป็น สถาปัตยกรรมซอฟต์แวร์ ABC_M1.2 ซึ่งได้จากการทำงานของเครื่องมือ สามารถอธิบายได้ว่า มีรายการข้อกำหนดความต้องการที่ได้รับผลกระทบ คือ รายการข้อกำหนดความต้องการที่ 10 - Use for personnel business workflow grouping ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ PersonnelInterface และรายการข้อกำหนดความต้องการที่ 11 - Use for inventory business workflow grouping ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ InventoryInterface เป็นต้น นอกจากนี้ การแสดงผลกระทบของรายงานยังสามารถแสดงผลในส่วนของคุณสมบัติเชิงคุณภาพและชุดคำสั่งที่ได้รับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ รายละเอียดดังกล่าวได้แสดงไว้ในบทที่ 5



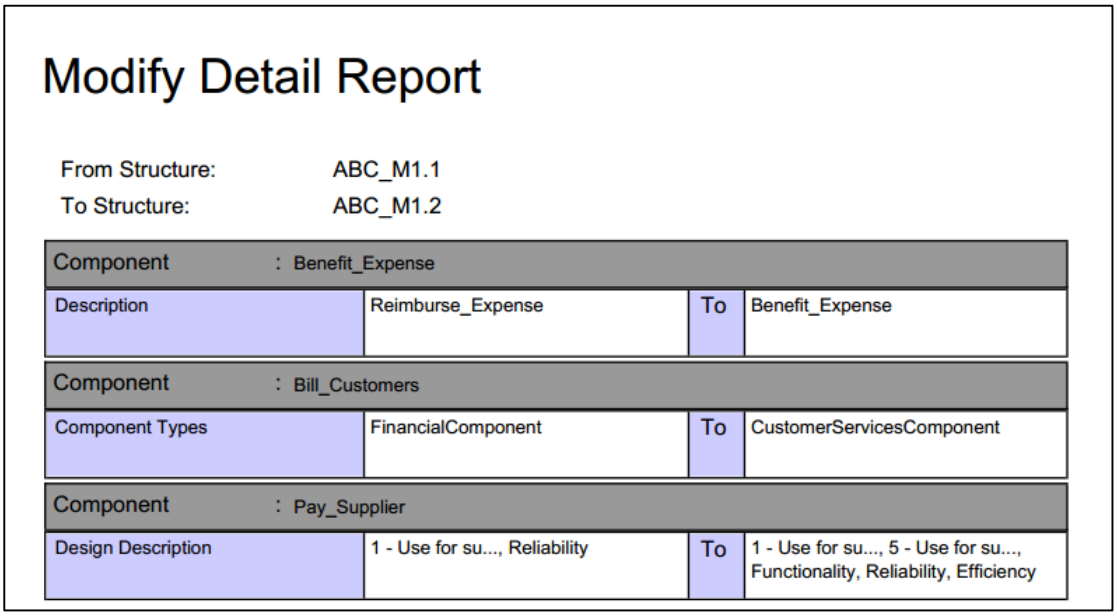
Change Report

From Structure: ABC_M1.1
To Structure: ABC_M1.2

No.	Element Name	Type	Difference Classification
1	InventoryInterface	Component	Add-In
2	PaymentInterface	Component	Add-In
3	PersonnelInterface	Component	Add-In
4	INF21	Interface	Add-In

รูปที่ 4.17 ตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลง
โครงสร้างสถาปัตยกรรมซอฟต์แวร์ในรูปแบบรายงาน

ตัวอย่างรายงานแสดงผลสรุปการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ จากสถาปัตยกรรมซอฟต์แวร์ ABC_M1.1 เป็น สถาปัตยกรรมซอฟต์แวร์ ABC_M1.2 ซึ่งได้จากการทำงานของเครื่องมือ สามารถอธิบายได้ว่า มีส่วนประกอบที่มีการออกแบบเพิ่มเติมจากสถาปัตยกรรมตั้งต้น คือ ส่วนประกอบ InventoryInterface, PaymentInterface และ PersonnelInterface และมีจุดเชื่อมต่อที่มีการออกแบบเพิ่มเติม ได้แก่ จุดเชื่อมต่อ INF21 เป็นต้น นอกจากนี้ การแสดงผลสรุปการเปลี่ยนแปลงของรายงานยังสามารถแสดงผลในส่วนของการแก้ไขพร้อมคำอธิบายการแก้ไขเบื้องต้น รวมถึงแสดงผลการลบส่วนประกอบจากการออกแบบสถาปัตยกรรมตั้งต้น รายละเอียดดังกล่าวได้แสดงไว้ในบทที่ 5



Modify Detail Report

From Structure: ABC_M1.1
To Structure: ABC_M1.2

Component	: Benefit_Expense		
Description	Reimburse_Expense	To	Benefit_Expense
Component	**: Bill_Customers**		
Component Types	FinancialComponent	To	CustomerServicesComponent
Component	**: Pay_Supplier**		
Design Description	1 - Use for su..., Reliability	To	1 - Use for su..., 5 - Use for su..., Functionality, Reliability, Efficiency

รูปที่ 4.18 ตัวอย่างผลลัพธ์รายละเอียดการเปลี่ยนแปลง
โครงสร้างสถาปัตยกรรมซอฟต์แวร์ในรูปแบบรายงาน

จากตัวอย่างรายงานแสดงรายละเอียดการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ จากสถาปัตยกรรมซอฟต์แวร์ ABC_M1.1 เป็น สถาปัตยกรรมซอฟต์แวร์ ABC_M1.2 ซึ่งได้จากการทำงานของเครื่องมือ สามารถอธิบายได้ว่า มีส่วนประกอบที่มีการแก้ไขการออกแบบจากสถาปัตยกรรมตั้งต้น คือ ส่วนประกอบ Benefit_Expense มีการแก้ไขคำอธิบาย จากเดิม คือ Reimburse_Expense เป็น Benefit_Expense, ส่วนประกอบ Bill_Customers มีการแก้ไขประเภทของส่วนประกอบจากเดิม คือ FinancialComponent เป็น CustomerServicesComponent และส่วนประกอบ Pay_Supplier มีการแก้ไขคำอธิบายการออกแบบจากเดิม กำหนดรายการข้อกำหนดความต้องการที่ 1 - Use for company account processing balance และคุณลักษณะเชิงคุณภาพด้านความน่าเชื่อถือ โดยมีการเพิ่มเติมรายการข้อกำหนดความต้องการที่ 5 - Use for print customer payment slip และกำหนดคุณลักษณะเชิงคุณภาพด้านหน้าที่การทำงาน และประสิทธิภาพเพิ่มเติมจากคำอธิบายการออกแบบเดิม เป็นต้น นอกจากนี้ การแสดงรายละเอียดการเปลี่ยนแปลงของรายงานยังสามารถแสดงผลในส่วนของรายละเอียดการแก้ไขส่วนประกอบจากการออกแบบสถาปัตยกรรมอื่นๆ รายละเอียดดังกล่าวได้แสดงไว้ในบทที่ 5

บทที่ 5
การทดสอบเครื่องมือ

การทดสอบเครื่องมือเป็นการทดสอบเพื่อตรวจสอบความถูกต้องของเครื่องมือที่พัฒนาขึ้น ซึ่งมีความสามารถในการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ ตลอดจนสรุปผลความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ โดยกล่าวถึงรายละเอียดของการทดสอบและผลของการทดสอบ มีรายละเอียดดังต่อไปนี้

5.1 การทดสอบเครื่องมือ

ผู้วิจัยได้ทำการแบ่งการทดสอบตามกรณีทดสอบ ทั้งหมดจำนวน 4 กรณีทดสอบ ซึ่งแยกตามลักษณะของกรณีทดสอบได้ ดังนี้

1. ตัวอย่างการเปลี่ยนแปลงของสถาปัตยกรรมซอฟต์แวร์ ซึ่งมีปรับปรุงเงื่อนไขสำหรับการตรวจจับผลกระทบจากการเปลี่ยนแปลงที่ได้กำหนดไว้ครบทุกเงื่อนไข 1 กรณีทดสอบ
2. ตัวอย่างการเปลี่ยนแปลงของสถาปัตยกรรมซอฟต์แวร์จากตัวอย่าง 3 กรณีทดสอบ

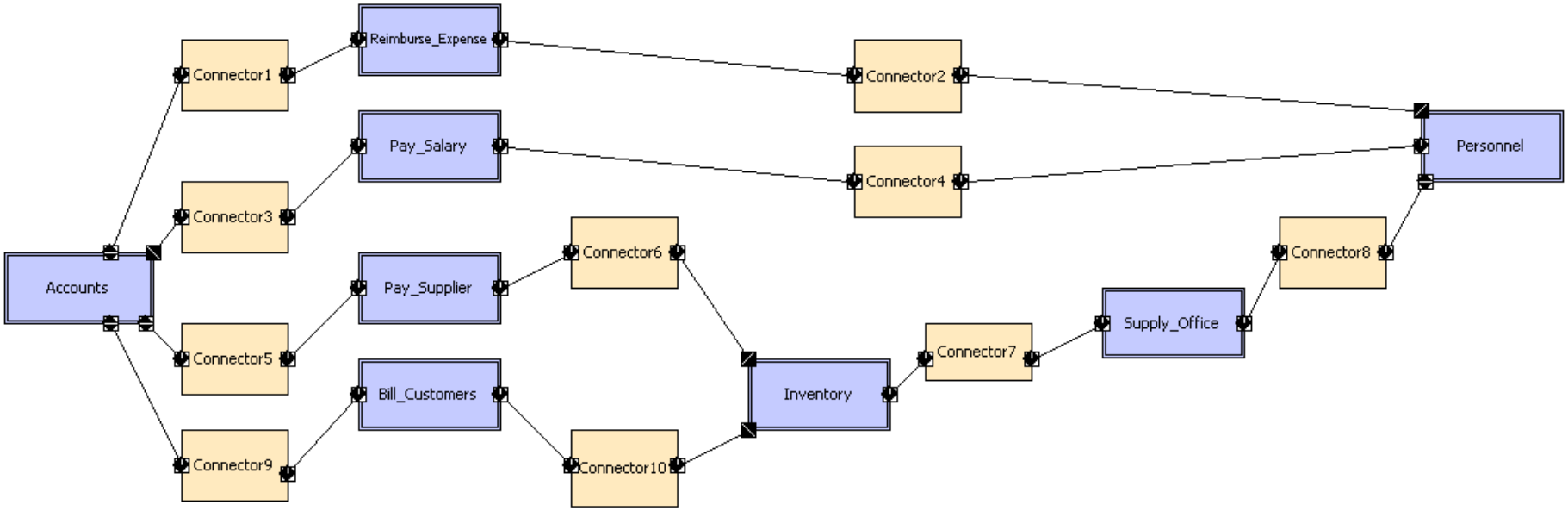
โดยกรณีทดสอบดังกล่าวนำมาจากงานวิจัย Evolution Styles: Foundations and Tool Support for Software Architecture Evolution [26] ได้แก่ ระบบบริหารงานบริษัทเอบีซี, ตัวอย่างระบบจาก Arcade Game Maker Pedagogical Product Line: Architecture Documentation, Volume 2 - Software Architecture Views [27] ได้แก่ ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ และตัวอย่างระบบจากหนังสือ Disciplined Agile Delivery (DAD) [28] ได้แก่ ระบบการลงทะเบียนมหาวิทยาลัย

กรณีทดสอบที่หนึ่ง ระบบบริหารงานบริษัทเอบีซี ได้มีการปรับปรุงการเปลี่ยนแปลงลักษณะโครงสร้างสถาปัตยกรรมซอฟต์แวร์ให้สอดคล้องและครอบคลุมกับความสามารถการทำงานของเครื่องมือที่ระบุไว้ในส่วนของขอบเขตงานวิจัยซึ่งได้กล่าวไว้ข้างต้น และกรณีทดสอบที่สอง เป็นการยกตัวอย่างจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ของระบบบริหารงานบริษัทเอบีซีโดยทั่วไป กรณีทดสอบที่สาม ใช้ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ และกรณีทดสอบที่สี่ ใช้ระบบการลงทะเบียนมหาวิทยาลัย นำมายกตัวอย่างเพื่อให้เข้าใจในการนำเครื่องมือไปประยุกต์ใช้งาน ผลที่ได้จากการทดสอบเป็นข้อมูลรายงานสรุปส่วนประกอบที่ได้รับ

ผลกระทบจากการเปลี่ยนแปลง รายงานสรุปผลความเปลี่ยนแปลง และรายงานแสดงรายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ โดยหลังจากทำการทดสอบด้วยข้อมูลของกรณีทดสอบทั้งหมด สามารถแสดงตัวอย่างผลการทดสอบ แยกตามกรณีทดสอบ ได้ดังนี้

5.1.1 **กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี (กำหนดเงื่อนไขการเปลี่ยนแปลงให้ครอบคลุมกับขอบเขตความสามารถในการตรวจจับของเครื่องมือ)**

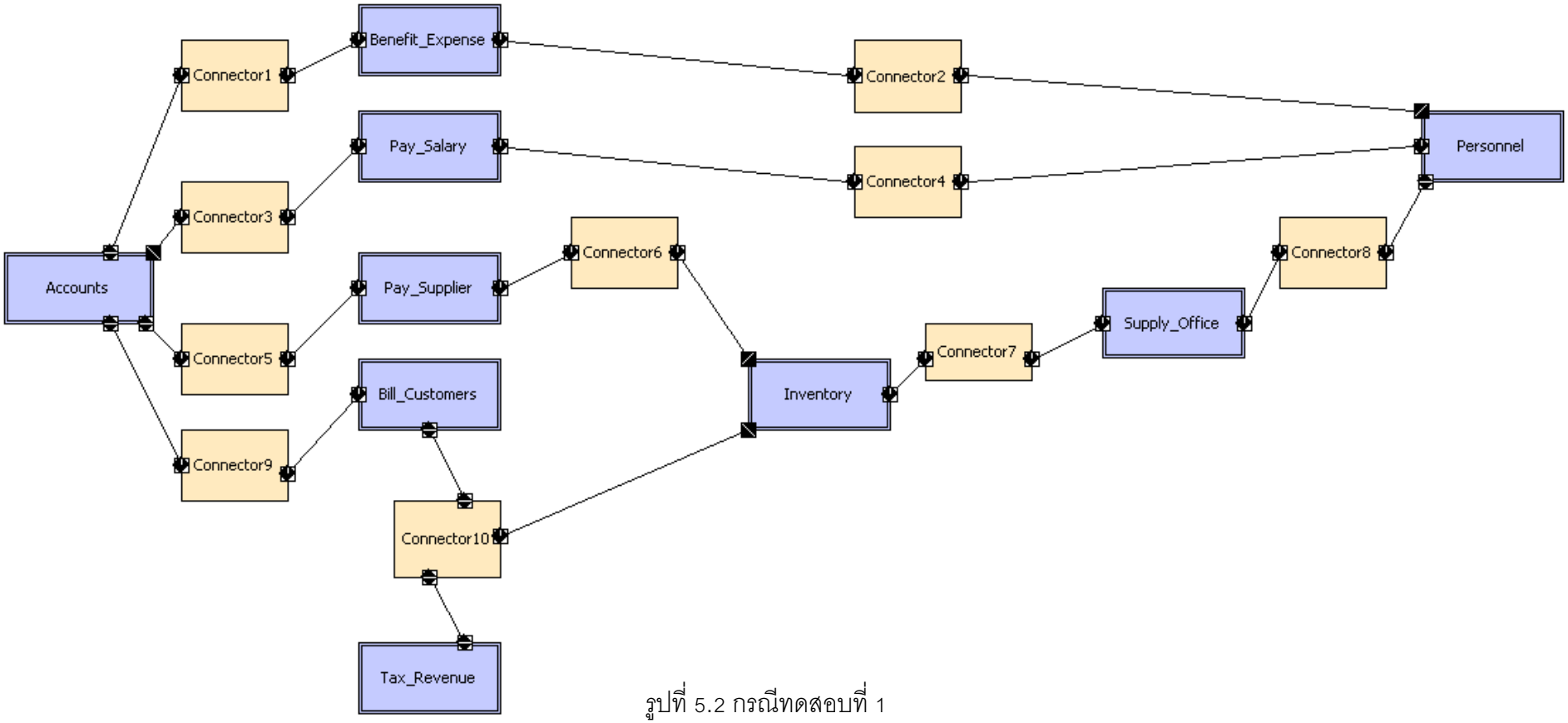
บริษัทเอบีซีต้องการที่จะปรับเปลี่ยนโครงสร้างสถาปัตยกรรมซอฟต์แวร์เดิมของบริษัท เพื่อให้สามารถรองรับการทำงานที่มีความจำเป็นต่อองค์กร ในส่วนงานของการจ่ายเงินคืนให้กับพนักงานภายใน การชำระภาษีให้กับลูกค้าของบริษัท รวมถึงการปรับเปลี่ยนรูปแบบการส่งผ่านข้อมูลด้านการเงินให้มีความปลอดภัยมากยิ่งขึ้น เนื่องจากโครงสร้างซอฟต์แวร์เดิมของบริษัทมีข้อจำกัดต่อการตอบสนองความต้องการที่เพิ่มเติมขึ้น ซึ่งมีผลต่อการแข่งขันในเชิงธุรกิจ อีกทั้งสถาปัตยกรรมซอฟต์แวร์เดิมของบริษัทยังผ่านการใช้งานมาเป็นระยะเวลานานตั้งแต่เริ่มก่อตั้งบริษัท จึงขาดการบำรุงรักษาให้สามารถทำงานได้อย่างมีประสิทธิภาพ และตอบสนองความต้องการดังที่ได้กล่าวมาข้างต้น เพื่อให้มีความสอดคล้องต่อสภาพการดำเนินธุรกิจในปัจจุบันขององค์กร ผู้บริหารของบริษัทเอบีซี จึงได้มีจัดการประชุมร่วมระหว่างผู้บริหารหน่วยงานภายในองค์กร ทีมงานฝ่ายเทคโนโลยีสารสนเทศ ผู้ปฏิบัติงาน ตลอดจนผู้ที่เกี่ยวข้องกับโครงการการปรับปรุงโครงสร้างซอฟต์แวร์ของบริษัท หลังจากการประชุมเสร็จสิ้น ที่ประชุมจึงได้ข้อสรุปร่วมกันในการปรับปรุงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ขององค์กร ให้รองรับความต้องการที่เพิ่มเติมขึ้น และเพื่อให้ซอฟต์แวร์มีความสามารถสนับสนุนการทำงานโดยมีความสอดคล้องกับความต้องการของผู้ปฏิบัติงานในปัจจุบัน โดยภาพรวมโครงสร้างสถาปัตยกรรมซอฟต์แวร์เดิมของบริษัทเอบีซี ก่อนการเปลี่ยนแปลง สามารถแสดงได้ดังรูปที่ 5.1



รูปที่ 5.1 กรณีทดสอบที่ 1

ภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงานบริษัทเอบีซี ก่อนการเปลี่ยนแปลง

เมื่อหัวหน้าทีมงานฝ่ายเทคโนโลยีสารสนเทศของบริษัทเอบีซีได้รับมอบหมายงาน ในการวิเคราะห์ รวบรวมความต้องการ และนำเสนอแนวทางการแก้ไขปัญหาที่เกี่ยวข้องกับโครงสร้างสถาปัตยกรรมซอฟต์แวร์เดิม หัวหน้าทีมงานและกลุ่มนักออกแบบสถาปัตยกรรมซอฟต์แวร์ จึงได้ทำการออกแบบโครงสร้างสถาปัตยกรรมซอฟต์แวร์ใหม่ ที่มีการปรับเปลี่ยนโครงสร้าง โดยเพิ่มเติมส่วนประกอบบางส่วน เพื่อทำให้ระบบซอฟต์แวร์ใหม่ขององค์กรมีความสามารถในการรองรับความต้องการของผู้ปฏิบัติงานและแก้ไขปัญหาที่เกิดขึ้นได้ ซึ่งมีรายละเอียดของการปรับปรุง ได้แก่ การปรับปรุงส่วนงานการจ่ายเงินคืนให้กับพนักงาน ซึ่งทำการเปลี่ยนแปลงคำอธิบายของส่วนงานให้มีความครอบคลุมสามารถรองรับในการจ่ายสิทธิประโยชน์ต่างๆ ให้กับพนักงานภายในองค์กร โดยมีความสอดคล้องกับระเบียบและข้อบังคับขององค์กร นอกจากนี้ ยังมีการปรับปรุงการเชื่อมต่อของส่วนงานการจ่ายเงินเดือนพนักงานให้มีความปลอดภัยเพิ่มมากยิ่งขึ้น นอกจากนี้ยังมีการเพิ่มเติมส่วนงานการจ่ายภาษีและออกใบชำระหนี้ให้กับลูกค้า ซึ่งการปรับเปลี่ยนโครงสร้างสถาปัตกรรมซอฟต์แวร์ใหม่ดังกล่าว มีความสามารถรองรับความต้องการในการปฏิบัติงานตลอดจนเพิ่มขีดความสามารถให้กับการดำเนินธุรกิจขององค์กรเพิ่มมากยิ่งขึ้น โดยภาพรวมโครงสร้างสถาปัตยกรรมซอฟต์แวร์ใหม่ของบริษัทเอบีซี หลังการเปลี่ยนแปลง สามารถแสดงได้ดังรูปที่ 5.2



รูปที่ 5.2 กรณีทดสอบที่ 1

ภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงานบริษัทเอบีซี หลังการเปลี่ยนแปลง

ทั้งนี้ ข้อมูลที่นักวิเคราะห์ระบบทำการจัดเก็บ รวบรวม และวิเคราะห์ความต้องการที่เกี่ยวข้องกับการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงานบริษัทเอบีซีและได้ใช้เครื่องมือที่พัฒนาขึ้นในการบันทึกข้อมูลความต้องการทั้งหมด เพื่อใช้เป็นข้อมูลสำหรับโครงการ

โดยตัวอย่างข้อมูลและผลการทดสอบเครื่องมือส่วนการจัดการคำอธิบายข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี สามารถแสดงตัวอย่างได้ ดังตารางที่ 5.1 - 5.5

ตารางที่ 5.1 ตัวอย่างและผลลัพธ์การบันทึกคำอธิบายโครงการ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M02

ระบบบริหารงานบริษัทเอบีซี				
คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์โครงการ	String	PK	สำเร็จ
code	รหัสโครงการ	String	ABC-CODE-M02	สำเร็จ
name	ชื่อโครงการ	String	ABC-M2	สำเร็จ
description	คำอธิบายโครงการ	String	Maintain software architecture for ABC?s adaptive business activities	สำเร็จ
requirementID	ค่าคีย์ของคลาส Requirement	List	-	-
fileName	ชื่อแฟ้มข้อมูลโครงการ	String	-	-

ตารางที่ 5.2 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ที่ 9

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์ความต้องการ	String	9	สำเร็จ
code	รหัสความต้องการ	String	ABC-REQ09	สำเร็จ
name	ชื่อความต้องการ	String	Use for payment business workflow grouping	สำเร็จ
description	คำอธิบายความต้องการ	String	-	สำเร็จ
status	สถานะความต้องการ	String	open	สำเร็จ
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U	สำเร็จ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U	สำเร็จ
submittedDate	วันที่สร้าง	Date	13/02/2013	สำเร็จ
updatedDate	วันที่ปรับปรุงข้อมูล	Date	13/02/2013	สำเร็จ

ตารางที่ 5.3 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ที่ 10

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์ความต้องการ	String	10	สำเร็จ
code	รหัสความต้องการ	String	ABC-REQ10	สำเร็จ
name	ชื่อความต้องการ	String	Use for personnel business workflow grouping	สำเร็จ
description	คำอธิบายความต้องการ	String	-	สำเร็จ

ตารางที่ 5.3 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ที่ 10 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
status	สถานะความต้องการ	String	open	สำเร็จ
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U	สำเร็จ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U	สำเร็จ
submittedDate	วันที่สร้าง	Date	13/02/2013	สำเร็จ
updatedDate	วันที่ปรับปรุงข้อมูล	Date	13/02/2013	สำเร็จ

ตารางที่ 5.4 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ที่ 11

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์ความต้องการ	String	11	สำเร็จ
code	รหัสความต้องการ	String	ABC-REQ11	สำเร็จ
name	ชื่อความต้องการ	String	Use for inventory business workflow grouping	สำเร็จ
description	คำอธิบายความต้องการ	String	-	สำเร็จ
status	สถานะความต้องการ	String	open	สำเร็จ
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U	สำเร็จ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U	สำเร็จ
submittedDate	วันที่สร้าง	Date	13/02/2013	สำเร็จ
updatedDate	วันที่ปรับปรุงข้อมูล	Date	13/02/2013	สำเร็จ

ทั้งนี้ รายละเอียดข้อมูลการทดสอบในส่วนการบันทึกข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซีทั้งหมดได้แสดงไว้ในภาคผนวก ก

เมื่อข้อมูลความต้องการและข้อมูลต่างๆ จากผู้เกี่ยวข้องกับโครงการได้ถูกกำหนดโดยนักวิเคราะห์ระบบเรียบร้อยแล้ว กลุ่มนักออกแบบจึงดำเนินการแก้ไขและปรับปรุงโครงสร้างสถาปัตยกรรมซอฟต์แวร์โดยใช้ข้อมูลดังกล่าวเป็นพื้นฐาน อีกทั้งยังได้มีการใช้เครื่องมือที่พัฒนาขึ้นในการบันทึกคำอธิบายการออกแบบให้กับส่วนประกอบต่างๆ ของโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่ปรับปรุงแก้ไขเพิ่มเติมอีกด้วย ตัวอย่างข้อมูลและผลการทดสอบเครื่องมือส่วนการระบุคำอธิบายการออกแบบให้กับส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงานบริษัทเอบีซี สามารถแสดงตัวอย่างได้ ดังตารางที่ 5.5

ตารางที่ 5.5 ตัวอย่างและผลการบันทึกคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M02

ระบบบริหารงานบริษัทเอบีซี				
ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ	ผลการบันทึกข้อมูล
1	Accounts	1 - Use for company account processing balance	Efficiency, Maintainability, Reliability, Portability	สำเร็จ
2	Benefit_Expense	2 - Use for payback expense and automatic accounting balance		สำเร็จ

ตารางที่ 5.5 ตัวอย่างและผลการบันทึกคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M02 (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ	ผลการบันทึกข้อมูล
3	Pay_Salary	3 - Use for salary payment and automatic accounting balance	Functionality	สำเร็จ
4	Pay_Supplier	4 - Use for supplier payment by export	Efficiency, Functionality, Reliability	สำเร็จ
5	Bill_Customers	5 - Use for print customer payment slip	Usability	สำเร็จ
6	Inventory	6 - Use for inventory automatic balance	Efficiency	สำเร็จ
7	Supply_Office	7 - Use for office stuff and automatic update current of office stock	Maintainability	สำเร็จ

ทั้งนี้ รายละเอียดข้อมูลการทดสอบในส่วนการบันทึกคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์โครงการระบบบริหารงานบริษัทเอบีซีทั้งหมดได้แสดงไว้ในภาคผนวก ข

หลังจากนักออกแบบสถาปัตยกรรมซอฟต์แวร์ได้ทำการปรับปรุงแก้ไขโครงสร้างสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงานบริษัทเอบีซีแล้วเสร็จ ทางหัวหน้าหน่วยงานเทคโนโลยีสารสนเทศ นักพัฒนาซอฟต์แวร์ของบริษัท และผู้ที่เกี่ยวข้องกับโครงการมีความต้องการทราบถึงผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่เกี่ยวข้องกับส่วนงาน หรือชุดคำสั่งของซอฟต์แวร์ปัจจุบันว่ามีผลกระทบในส่วนใดบ้าง สถาปนิกซอฟต์แวร์จึงได้ใช้เครื่องมือที่พัฒนาขึ้นในการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ เพื่อนำข้อมูลที่ได้จากการประมวลผลของเครื่องมือไปใช้รายงานกับผู้ที่เกี่ยวข้องกับโครงการต่อไป

โดยตัวอย่างผลลัพธ์การทดสอบเครื่องมือส่วนการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงานบริษัทเอบีซี สามารถแสดงตัวอย่างได้ ดังตารางที่ 5.6 - 5.10

ตารางที่ 5.6 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อความต้องการซอฟต์แวร์

ผลกระทบต่อความต้องการซอฟต์แวร์		
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2		
ลำดับที่	ชื่อความต้องการ	ส่วนประกอบที่เกี่ยวข้อง
1	1 - Use for company account processing balance	Accounts
2	2 - Use for payback expense and automatic accounting balance	Benefit_Expense
3	3 - Use for salary payment and automatic accounting balance	Pay_Salary
4	4 - Use for supplier payment by export	Pay_Supplier
5	5 - Use for print customer payment slip	Bill_Customers

จากตัวอย่างผลลัพธ์การตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ ในส่วนผลกระทบต่อความต้องการซอฟต์แวร์ สามารถอธิบายได้ว่า มีรายการข้อกำหนดความต้องการที่ได้รับผลกระทบ คือ รายการข้อกำหนดความต้องการที่ 1 - Use for company account processing balance ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ Accounts

และรายการข้อกำหนดความต้องการที่ 2 - Use for payback expense and automatic accounting balance ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ Benefit_Expense เป็นต้น

ตารางที่ 5.7 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ

ผลกระทบต่อคุณลักษณะเชิงคุณภาพ		
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2		
ลำดับที่	ชื่อคุณลักษณะเชิงคุณภาพ	ส่วนประกอบที่เกี่ยวข้อง
1	Efficiency	Accounts, Inventory, Pay_Supplier
2	Functionality	Pay_Salary, Pay_Supplier
3	Maintainability	Accounts, Supply_Office
4	Portability	Accounts, Personnel
5	Reliability	Accounts, Pay_Supplier

จากตัวอย่างผลลัพธ์การตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ ในส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ สามารถอธิบายได้ว่า มีคุณลักษณะเชิงคุณภาพที่ได้รับผลกระทบ คือ คุณลักษณะเชิงคุณภาพด้านประสิทธิภาพ ด้านหน้าที่การทำงาน ด้านการบำรุงรักษา ด้านความสะดวกในการพกพา และด้านความน่าเชื่อถือ ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ Accounts, Inventory, Pay_Supplier, Pay_Salary, Supply_Office และ Personnel ตามลำดับ

ตารางที่ 5.8 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อชุดคำสั่ง

ผลกระทบต่อชุดคำสั่ง		
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2		
ลำดับที่	ชื่อชุดคำสั่ง	ส่วนประกอบที่เกี่ยวข้อง
1	com.abc.customerSevicesImpl	Bill_Customers

ตารางที่ 5.8 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อชุดคำสั่ง (ต่อ)

ลำดับที่	ชื่อชุดคำสั่ง	ส่วนประกอบที่เกี่ยวข้อง
2	com.abc.financialImpl	Accounts, Benefit_Expense, Pay_Salary, Pay_Supplier
3	com.abc.HumanResourceImpl	Personnel

จากตัวอย่างผลลัพธ์การตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ ในส่วนผลกระทบต่อชุดคำสั่ง สามารถอธิบายได้ว่า มีชุดคำสั่งที่ได้รับผลกระทบ คือ ชุดคำสั่ง com.abc.customerSevicesImpl ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ Bill_Customers ชุดคำสั่ง com.abc.financialImpl ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ Accounts, Benefit_Expense, Pay_Salary, Pay_Supplier และชุดคำสั่ง com.abc.HumanResourceImpl ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ Personnel ตามลำดับ

ตารางที่ 5.9 ตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี

ระบบบริหารงานบริษัทเอบีซี			
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1			
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2			
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	ประเภทความเปลี่ยนแปลง
1	Tax_Revenue	Component	Add-In
2	INF21	Interface	Add-In
3	INF-C10-3	Interface	Add-In
4	Link17	Link	Add-In
5	Benefit_Expense	Component	Modify

จากตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ สามารถอธิบายได้ว่า มีส่วนประกอบที่มีการออกแบบเพิ่มเติมขึ้นมา จากสถาปัตยกรรมตั้งต้น ได้แก่ ส่วนประกอบ Tax_Revenue และมีการออกแบบจุดเชื่อมต่อเพิ่มเติม

คือ จุดเชื่อมต่อ INF21, INF-C10-3 มีการออกแบบเส้นเชื่อมเพิ่มเติม คือ เส้นเชื่อม Link17 นอกจากนี้ส่วนประกอบที่มีการแก้ไข คือ ส่วนประกอบ Benefit_Expense เป็นต้น

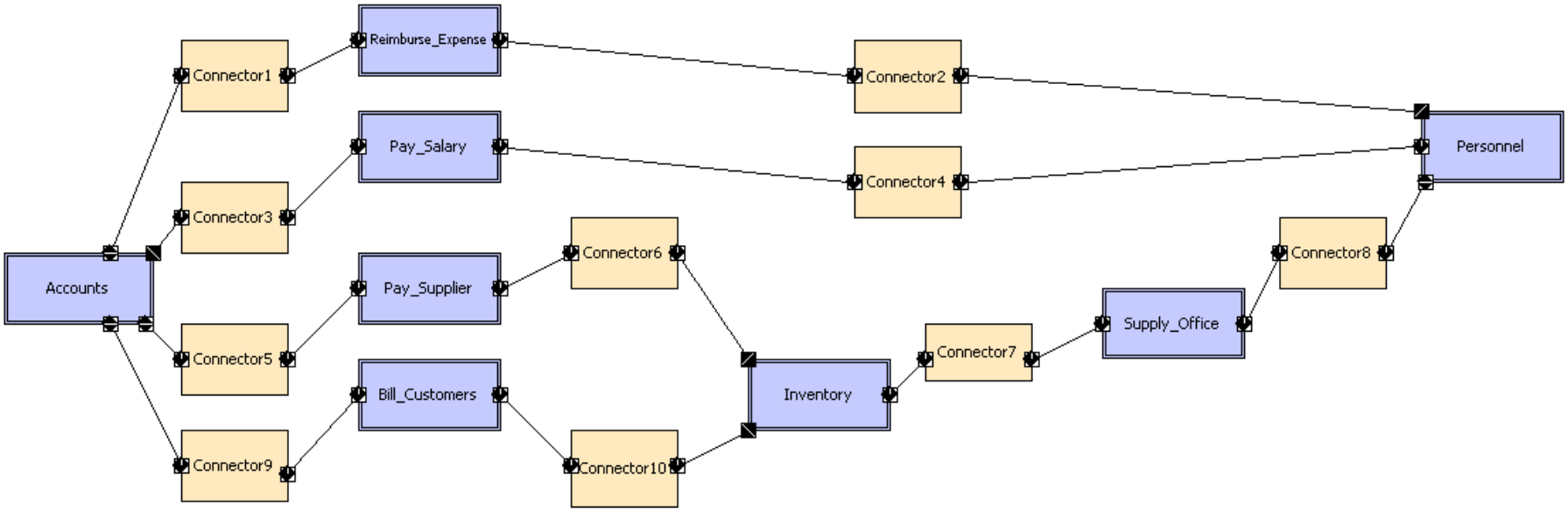
ตารางที่ 5.10 ตัวอย่างการตรวจจับรายละเอียดการเปลี่ยนแปลงส่วนประกอบ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี

ระบบบริหารงานบริษัทเอบีซี					
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1					
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2					
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	การเปลี่ยนแปลงคุณสมบัติ	ค่าเดิม	ค่าใหม่
1	Benefit_Expense	Component	Description	Reimburse_Expense	Benefit_Expense
2	Bill_Customers	Component	Component Types	Financial Component	Customer Services Component
3	Pay_Supplier	Component	Design Description	Reliability	Functionality, Reliability, Efficiency
4	Connector10	Connector	No of Interfaces	2	3
5	Connector4	Connector	Connector Types	TCPIP Connector	IPTunnels Connector

จากตัวอย่างผลลัพธ์การตรวจจับรายละเอียดการเปลี่ยนแปลงส่วนประกอบ ที่ได้จากการทำงานของเครื่องมือ สามารถอธิบายได้ว่า มีส่วนประกอบที่มีการแก้ไขรายละเอียดจากสถาปัตยกรรมตั้งต้น ได้แก่ ส่วนประกอบ Benefit_Expense โดยมีการแก้ไขคำอธิบาย จากเดิม คือ Reimburse_Expense เปลี่ยนเป็น Benefit_Expense และส่วนประกอบ Bill_Customers มีการแก้ไขประเภทของส่วนประกอบ จากเดิม คือ FinancialComponent เปลี่ยนเป็น CustomerServicesComponent เป็นต้น โดยรายละเอียดผลการทดสอบทั้งหมดได้แสดงไว้ในภาคผนวก ค

5.1.2 กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี

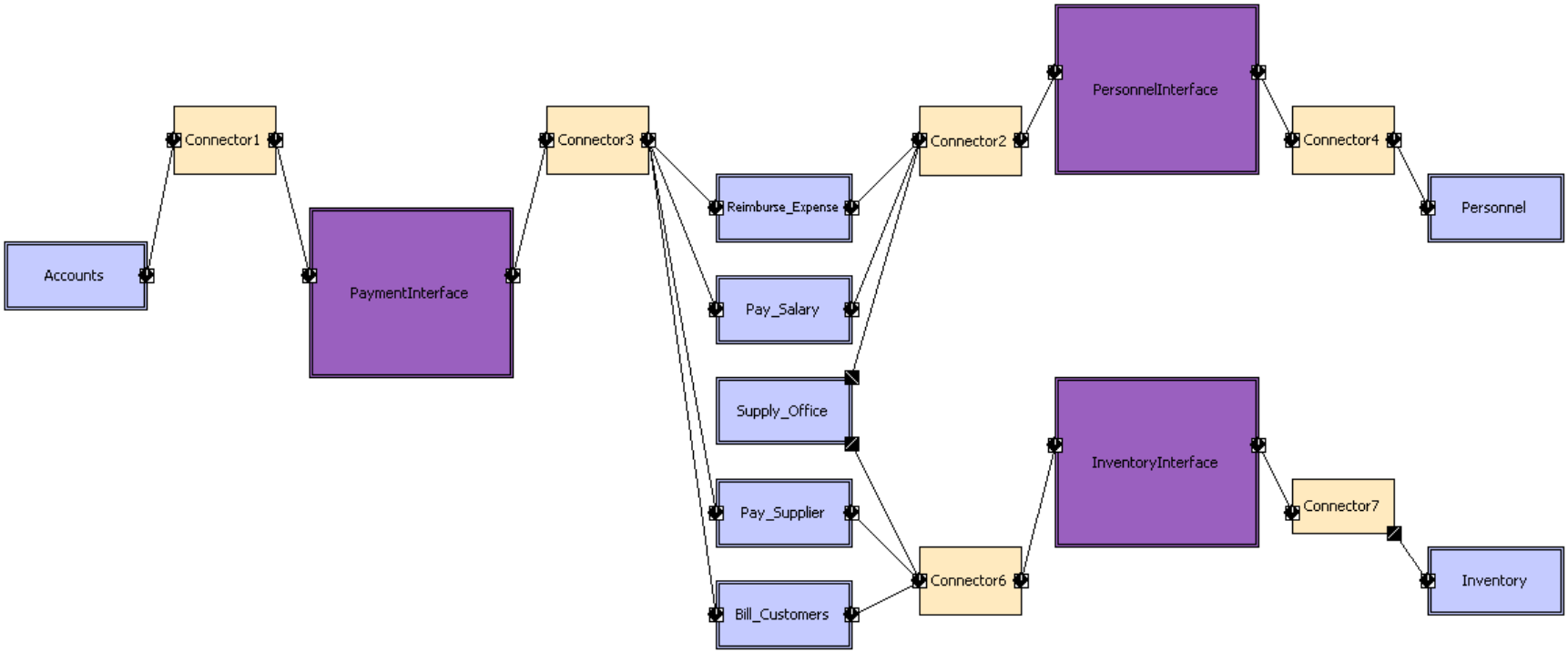
บริษัทเอบีซีต้องการปรับปรุงโครงสร้างซอฟต์แวร์เดิมของบริษัท เพื่อรองรับการขยายตัวทางธุรกิจที่มีการแข่งขันเพิ่มมากขึ้น โดยสถาปัตยกรรมซอฟต์แวร์เดิมมีลักษณะแยกการทำงานการเชื่อมต่อกันระหว่างระบบต่างๆ ภายในองค์กรอย่างอิสระ ยกตัวอย่างเช่น ระบบบัญชีมีความสามารถในการจ่ายเงินคืน การจ่ายเงินเดือน การจ่ายเงินให้กับผู้ผลิต โดยเป็นการติดต่อกันสองฝั่งระหว่างระบบต่อระบบ เป็นต้น ซึ่งผู้บริหารและหัวหน้าหน่วยงานเทคโนโลยีสารสนเทศ ได้ทำการจัดการประชุมเพื่อวิเคราะห์ผลเสียและข้อจำกัดของรูปแบบสถาปัตยกรรมลักษณะดังกล่าว พบว่าโครงสร้างสถาปัตยกรรมซอฟต์แวร์ในลักษณะเดิมมีปัญหาหลายด้าน ยกตัวอย่างเช่น การที่แต่ละระบบติดต่อและพัฒนาแยกกันในลักษณะที่มีความเป็นอิสระ ทำให้อาจจะเกิดความผิดพลาดของการส่งผ่านข้อมูลระหว่างระบบได้ จากการที่ระบบดังกล่าวมีการพัฒนาแยกกันโดยเฉพาะนั้น ทำให้บริษัทต้องใช้ทรัพยากรและต้นทุนจำนวนมากในการบำรุงรักษาซอฟต์แวร์ในแต่ละระบบให้ยังสามารถทำงานต่อไปได้ อีกทั้งการปรับปรุงแก้ไขระบบต่างๆ เพิ่มเติมในอนาคตจะทำได้ยากอีกด้วย โดยหลังจากการประชุมเสร็จสิ้นลงที่ประชุมจึงมีข้อสรุปโดยเห็นตรงกันว่าข้อจำกัดของโครงสร้างซอฟต์แวร์ในลักษณะดังกล่าวส่งผลเสียต่อการแข่งขันและการดำเนินธุรกิจของบริษัทในอนาคต ทำให้ผู้บริหารบริษัทเอบีซีมองเห็นความจำเป็นต่อการปรับปรุงแก้ไขโครงสร้างซอฟต์แวร์เดิม ผู้บริหารบริษัทเอบีซีจึงมีการมอบหมายให้ฝ่ายเทคโนโลยีสารสนเทศเริ่มต้นกระบวนการปรับเปลี่ยนรูปแบบโครงสร้างสถาปัตยกรรมซอฟต์แวร์ เพื่อแก้ไขปัญหาที่เกิดขึ้นและทำให้โครงสร้างสถาปัตยกรรมซอฟต์แวร์มีความเหมาะสมกับการดำเนินธุรกิจขององค์กรในปัจจุบัน โดยภาพรวมสถาปัตยกรรมซอฟต์แวร์บริษัทเอบีซี ก่อนการเปลี่ยนแปลงสามารถแสดงได้ดังรูปที่ 5.3



รูปที่ 5.3 กรณีทดสอบที่ 2

ภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงานบริษัทเอบีซี ก่อนการเปลี่ยนแปลง

จากปัญหาของรูปแบบโครงสร้างสถาปัตยกรรมซอฟต์แวร์เดิมที่ทางผู้บริหารบริษัทเอบีซีและหัวหน้าหน่วยงานเทคโนโลยีสารสนเทศได้วิเคราะห์ร่วมกัน หัวหน้าหน่วยงานเทคโนโลยีสารสนเทศจึงได้มอบหมายงานการออกแบบสถาปัตยกรรมซอฟต์แวร์ที่มีความเหมาะสมกับองค์กรและมีความยืดหยุ่นต่อการแข่งขันทางธุรกิจให้กับกลุ่มทีมงานนักออกแบบ หลังจากผ่านช่วงของการจัดเก็บรวบรวมข้อมูลความต้องการ การวิเคราะห์ผลดี ผลเสีย การประเมินลักษณะของการพัฒนาซอฟต์แวร์ รวมถึงวิเคราะห์ข้อมูลที่เกี่ยวข้องกับเทคโนโลยีและทรัพยากรต่างๆ ขององค์กรที่สามารถรองรับและสนับสนุนการปรับปรุงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ใหม่แล้ว ทางทีมงานนักออกแบบสถาปัตยกรรมซอฟต์แวร์ จึงได้สรุปผลการออกแบบโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่มีการนำรูปแบบการออกแบบในลักษณะของอินเตอร์เฟสอแดปเตอร์ สำหรับการเชื่อมต่อกันของระบบย่อยที่มีอยู่เดิมภายในองค์กรมาใช้งาน ซึ่งมีประโยชน์ทำให้เกิดความยืดหยุ่นต่อการพัฒนาและง่ายต่อการแก้ไขปรับปรุงส่วนต่อขยายเพิ่มเติมของซอฟต์แวร์ในอนาคตมากขึ้น โดยภาพรวมสถาปัตยกรรมซอฟต์แวร์บริษัทเอบีซี หลังการเปลี่ยนแปลง สามารถแสดงได้ดังรูปที่ 5.4



รูปที่ 5.4 กรณีทดสอบที่ 2

ภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงานบริษัทเอบีซี หลังการเปลี่ยนแปลง

โดยข้อมูลที่นักวิเคราะห์ระบบทำการจัดเก็บ รวบรวม และวิเคราะห์ความต้องการที่เกี่ยวข้องกับการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงานบริษัทเอบีซีและได้ใช้เครื่องมือที่พัฒนาขึ้นในการบันทึกข้อมูลความต้องการทั้งหมด เพื่อใช้เป็นข้อมูลสำหรับโครงการ

โดยตัวอย่างข้อมูลและผลการทดสอบเครื่องมือส่วนการจัดการคำอธิบายข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี สามารถแสดงตัวอย่างได้ ดังตารางที่ 5.11 - 5.14

ตารางที่ 5.11 ตัวอย่างและผลลัพธ์การบันทึกคำอธิบายโครงการ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M02

ระบบบริหารงานบริษัทเอบีซี				
คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์โครงการ	String	PK	สำเร็จ
code	รหัสโครงการ	String	ABC-CODE-M02	สำเร็จ
name	ชื่อโครงการ	String	ABC-M2	สำเร็จ
description	คำอธิบายโครงการ	String	Maintain software architecture for ABC?s adaptive business activities	สำเร็จ
requirementID	ค่าคีย์ของคลาส Requirement	List	-	-
fileName	ชื่อแฟ้มข้อมูลโครงการ	String	-	-

ตารางที่ 5.12 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ที่ 9

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์ความต้องการ	String	9	สำเร็จ
code	รหัสความต้องการ	String	ABC-REQ09	สำเร็จ
name	ชื่อความต้องการ	String	Use for payment business workflow grouping	สำเร็จ
description	คำอธิบายความต้องการ	String	-	สำเร็จ
status	สถานะความต้องการ	String	open	สำเร็จ
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U	สำเร็จ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U	สำเร็จ
submittedDate	วันที่สร้าง	Date	13/02/2013	สำเร็จ
updatedDate	วันที่ปรับปรุงข้อมูล	Date	13/02/2013	สำเร็จ

ตารางที่ 5.13 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ที่ 10

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์ความต้องการ	String	10	สำเร็จ
code	รหัสความต้องการ	String	ABC-REQ10	สำเร็จ

ตารางที่ 5.13 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ที่ 10 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
name	ชื่อความต้องการ	String	Use for personnel business workflow grouping	สำเร็จ
description	คำอธิบายความต้องการ	String	-	สำเร็จ
status	สถานะความต้องการ	String	open	สำเร็จ
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U	สำเร็จ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U	สำเร็จ
submittedDate	วันที่สร้าง	Date	13/02/2013	สำเร็จ
updatedDate	วันที่ปรับปรุงข้อมูล	Date	13/02/2013	สำเร็จ

ตารางที่ 5.14 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ที่ 11

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์ความต้องการ	String	11	สำเร็จ
code	รหัสความต้องการ	String	ABC-REQ11	สำเร็จ

ตารางที่ 5.14 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ที่ 11 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
name	ชื่อความต้องการ	String	Use for inventory business workflow grouping	สำเร็จ
description	คำอธิบายความต้องการ	String	-	สำเร็จ
status	สถานะความต้องการ	String	open	สำเร็จ
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U	สำเร็จ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U	สำเร็จ
submittedDate	วันที่สร้าง	Date	13/02/2013	สำเร็จ
updatedDate	วันที่ปรับปรุงข้อมูล	Date	13/02/2013	สำเร็จ

ทั้งนี้ รายละเอียดข้อมูลการทดสอบในส่วนการบันทึกข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซีทั้งหมดได้แสดงไว้ในภาคผนวก ก

นอกจากนี้ ในระหว่างการออกแบบเพื่อทำการปรับปรุงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ กลุ่มนักออกแบบได้ใช้ข้อมูลข้อกำหนดความต้องการดังกล่าวเป็นพื้นฐาน อีกทั้งยังได้มีการใช้เครื่องมือที่พัฒนาขึ้นในการบันทึกคำอธิบายการออกแบบให้กับส่วนประกอบต่างๆ ของโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่ปรับปรุงแก้ไขเพิ่มเติมอีกด้วย ตัวอย่างข้อมูลและผลการทดสอบเครื่องมือส่วนการระบุคำอธิบายการออกแบบให้กับส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงานบริษัทเอบีซี สามารถแสดงตัวอย่างได้ ดังตารางที่ 5.15

ตารางที่ 5.15 ตัวอย่างและผลการบันทึกคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M02

ระบบบริหารงานบริษัทเอบีซี				
ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ	ผลการบันทึกข้อมูล
1	Accounts	1 - Use for company account processing balance	Efficiency, Maintainability, Reliability, Portability	สำเร็จ
2	Reimburse_ Expense	2 - Use for payback expense and automatic accounting balance		สำเร็จ
3	Pay_Salary	3 - Use for salary payment and automatic accounting balance	Functionality	สำเร็จ
4	Pay_Supplier	4 - Use for supplier payment by export	Reliability	สำเร็จ
5	Bill_Customers	5 - Use for print customer payment slip	Usability	สำเร็จ

ทั้งนี้ รายละเอียดข้อมูลการทดสอบในส่วนการบันทึกคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์โครงการระบบบริหารงานบริษัทเอบีซีทั้งหมดได้แสดงไว้ในภาคผนวก ข

หลังจากนักออกแบบสถาปัตยกรรมซอฟต์แวร์ได้ทำการปรับปรุงแก้ไขโครงสร้างสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงานบริษัทเอบีซีแล้วเสร็จ ทางหัวหน้าหน่วยงานเทคโนโลยีสารสนเทศ นักพัฒนาซอฟต์แวร์ของบริษัท และผู้ที่เกี่ยวข้องกับโครงการมีความต้องการทราบถึงผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่เกี่ยวข้องกับส่วนงาน หรือ

ชุดคำสั่งของซอฟต์แวร์ปัจจุบันว่ามีผลกระทบในส่วนใดบ้าง สถาปนิกซอฟต์แวร์จึงได้ใช้เครื่องมือที่พัฒนาขึ้นในการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ เพื่อนำข้อมูลที่ได้จากการประมวลผลของเครื่องมือไปใช้รายงานกับผู้ที่เกี่ยวข้องกับโครงการต่อไป

โดยตัวอย่างผลลัพธ์การทดสอบเครื่องมือส่วนการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ระบบบริหารงานบริษัทเอบีซี สามารถแสดงตัวอย่างได้ ดังตารางที่ 5.16 - 5.20

ตารางที่ 5.16 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อความต้องการซอฟต์แวร์

ผลกระทบต่อความต้องการซอฟต์แวร์		
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2		
ลำดับที่	ชื่อความต้องการ	ส่วนประกอบที่เกี่ยวข้อง
1	10 - Use for personnel business workflow grouping	PersonnelInterface
2	11 - Use for inventory business workflow grouping	InventoryInterface
3	1 - Use for company account processing balance	Accounts
4	2 - Use for payback expense and automatic accounting balance	Reimburse_Expense
5	3 - Use for salary payment and automatic accounting balance	Pay_Salary

จากตัวอย่างผลลัพธ์การตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ ในส่วนผลกระทบต่อความต้องการซอฟต์แวร์ สามารถอธิบายได้ว่า มีรายการข้อกำหนดความต้องการที่ได้รับผลกระทบ คือ รายการข้อกำหนดความต้องการที่ 10 - Use for personnel business workflow grouping ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ PersonnelInterface และรายการข้อกำหนดความต้องการที่ 11 - Use for inventory business workflow grouping ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ InventoryInterface เป็นต้น

ตารางที่ 5.17 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ

ผลกระทบต่อคุณลักษณะเชิงคุณภาพ		
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2		
ลำดับที่	ชื่อคุณลักษณะเชิงคุณภาพ	ส่วนประกอบที่เกี่ยวข้อง
1	Efficiency	Accounts, Inventory
2	Functionality	Pay_Salary
3	Maintainability	Accounts, InventoryInterface, PaymentInterface, PersonnelInterface, Supply_Office
4	Portability	Accounts, Personnel
5	Reliability	Accounts, Pay_Supplier

จากตัวอย่างผลลัพธ์การตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ ในส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ สามารถอธิบายได้ว่า มีคุณลักษณะเชิงคุณภาพที่ได้รับผลกระทบ คือ คุณลักษณะเชิงคุณภาพด้านประสิทธิภาพ ด้านหน้าที่การทำงาน ด้านการบำรุงรักษา ด้านความสะดวกในการพกพา และด้านความน่าเชื่อถือ ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ Accounts, Inventory, Pay_Supplier, Pay_Salary, InventoryInterface, PaymentInterface, PersonnelInterface, Supply_Office และ Personnel ตามลำดับ

ตารางที่ 5.18 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อชุดคำสั่ง

ผลกระทบต่อชุดคำสั่ง		
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2		
ลำดับที่	ชื่อชุดคำสั่ง	ส่วนประกอบที่เกี่ยวข้อง
1	com.abc.financialImpl	Accounts, Bill_Customers, Pay_Salary, PaymentInterface, Reimburse_Expense, Pay_Supplier
2	com.abc.HumanResourceImpl	PersonnelInterface, Personnel
3	com.abc.InventoryImpl	Inventory, InventoryInterface, Supply_Office

จากตัวอย่างผลลัพธ์การตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ ในส่วนผลกระทบต่อชุดคำสั่ง สามารถอธิบายได้ว่า มีชุดคำสั่งที่ได้รับผลกระทบ คือ com.abc.financialImpl ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ Accounts, Bill_Customers, Pay_Salary, PaymentInterface, Reimburse_Expense, Pay_Supplier และชุดคำสั่ง com.abc.HumanResourceImpl ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ PersonnelInterface, Personnel และชุดคำสั่ง com.abc.InventoryImpl ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ Inventory, InventoryInterface, Supply_Office ตามลำดับ

ตารางที่ 5.19 ตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี

ระบบบริหารงานบริษัทเอบีซี			
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1			
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2			
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	ประเภทความเปลี่ยนแปลง
1	InventoryInterface	Component	Add-In

ตารางที่ 5.19 ตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	ประเภทความเปลี่ยนแปลง
2	PaymentInterface	Component	Add-In
3	PersonnelInterface	Component	Add-In
4	INF21	Interface	Add-In
5	INF22	Interface	Add-In

จากตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ สามารถอธิบายได้ว่า มีส่วนประกอบที่มีการออกแบบเพิ่มเติมขึ้นมา จากสถาปัตยกรรมตั้งต้น ได้แก่ ส่วนประกอบ InventoryInterface, PaymentInterface, PersonnelInterface และมีการออกแบบจุดเชื่อมต่อเพิ่มเติม คือ จุดเชื่อมต่อ INF21, INF22 เป็นต้น

ตารางที่ 5.20 ตัวอย่างการตรวจจับรายละเอียดการเปลี่ยนแปลงส่วนประกอบ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี

ระบบบริหารงานบริษัทเอบีซี					
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1					
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2					
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	การเปลี่ยนแปลงคุณสมบัติ	ค่าเดิม	ค่าใหม่
1	Accounts	Component	No of Interface	4	1
2	Inventory	Component	No of Interface	3	1

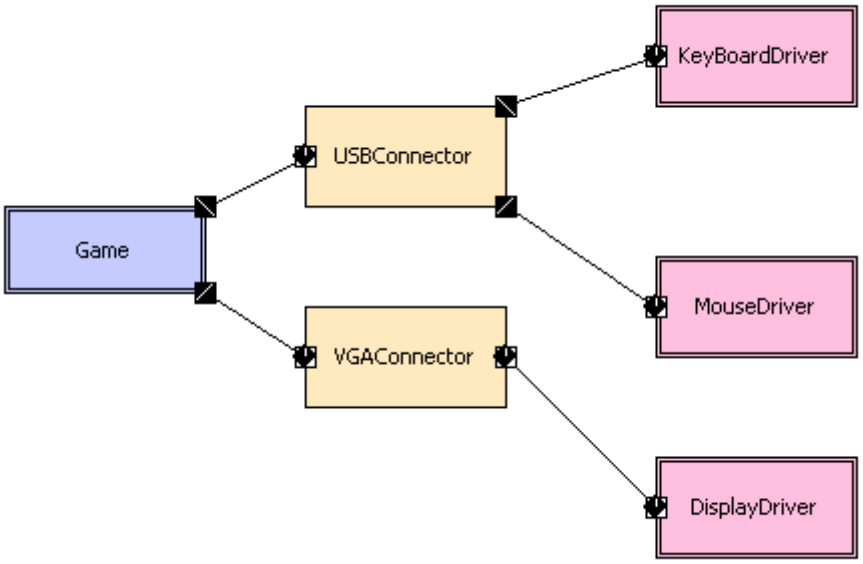
ตารางที่ 5.20 ตัวอย่างการตรวจจับรายละเอียดการเปลี่ยนแปลงส่วนประกอบ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	การเปลี่ยนแปลงคุณสมบัติ	ค่าเดิม	ค่าใหม่
3	Personnel	Component	No of Interface	3	1
4	Link10	Link	Link Pointer	INF-C5-2	INF-C3-2
5	Link12	Link	Link Pointer	INF13	INF25

จากตัวอย่างผลลัพธ์การตรวจจับรายละเอียดการเปลี่ยนแปลงส่วนประกอบ ที่ได้จากการทำงานของเครื่องมือ สามารถอธิบายได้ว่า มีส่วนประกอบที่มีการแก้ไขรายละเอียดจากสถาปัตยกรรมตั้งต้น ได้แก่ ส่วนประกอบ Accounts โดยมีการแก้ไขจำนวนจุดเชื่อมต่อ จากเดิม คือ 4 จุด เปลี่ยนเป็น 1 จุด และส่วนประกอบ Inventory มีการแก้ไขจำนวนจุดเชื่อมต่อ จากเดิม คือ 3 จุด เปลี่ยนเป็น 1 จุด เป็นต้น โดยรายละเอียดผลการทดสอบทั้งหมดได้แสดงไว้ในภาคผนวก ค

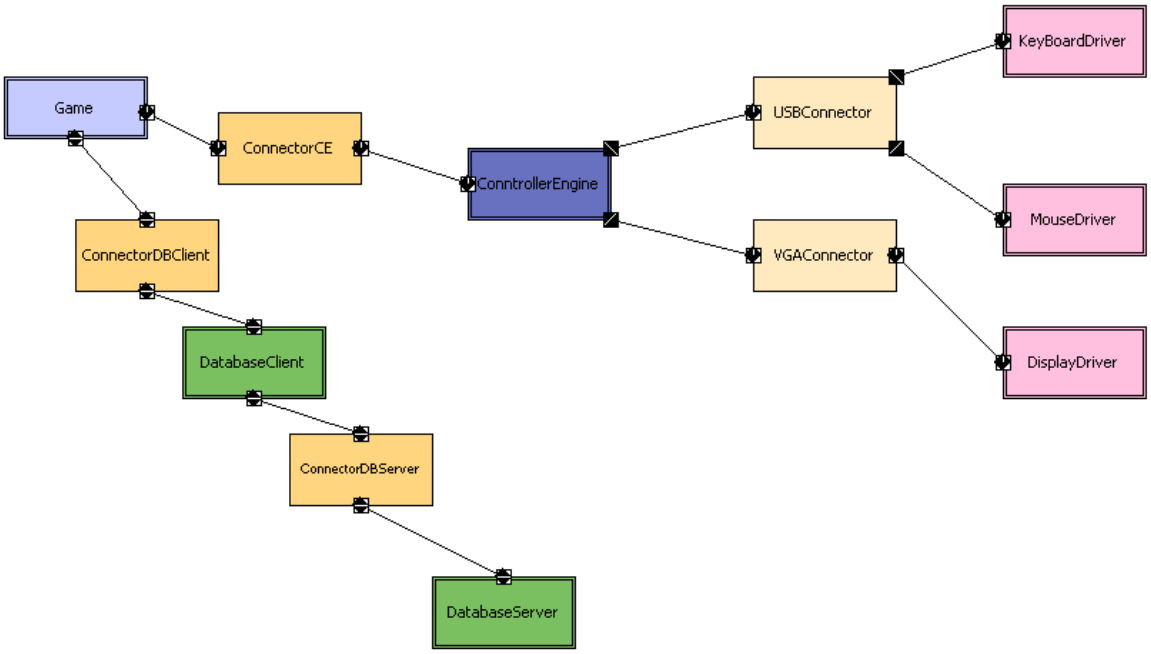
5.1.3 กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์

บริษัทผลิตเกมส์แห่งหนึ่ง ได้พัฒนาเกมส์เป็นลักษณะรูปแบบที่ใช้กับเครื่องเล่นเกมส์โดยเฉพาะ ซึ่งทำการวางจำหน่ายและได้รับการตอบรับจากกลุ่มนักเล่นเกมส์เป็นอย่างดี บริษัทผู้ผลิตจึงมีความต้องการในการปรับเปลี่ยนรูปแบบโครงสร้างสถาปัตยกรรมซอฟต์แวร์เดิมของเกมส์ ที่มีการวางจำหน่ายไปแล้วให้ครอบคลุมกับนักเล่นเกมส์ที่เป็นกลุ่มเป้าหมายใหม่ เพื่อขยายขอบเขตของตลาดการจัดจำหน่ายเกมส์ให้มากยิ่งขึ้น โดยมีการนำรูปแบบโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่เป็นลักษณะเกมส์ออนไลน์ ซึ่งมีข้อดีที่สำคัญ คือ สามารถเพิ่มการขยายตัวของสังคมผู้เล่นเกมส์ได้มากยิ่งขึ้นในวงกว้าง อีกทั้ง ยังช่วยลดต้นทุนในการผลิตและการจัดจำหน่าย ทำให้การจัดจำหน่ายเกมส์ไปยังกลุ่มนักเล่นเกมส์เป็นไปอย่างมีประสิทธิภาพมากยิ่งขึ้น บริษัทผู้ผลิตเกมส์จึงได้มีการว่าจ้างที่ปรึกษา ซึ่งมีความเชี่ยวชาญในการออกแบบสถาปัตยกรรมซอฟต์แวร์ลักษณะเกมส์ออนไลน์ เพื่อมาช่วยปรับปรุงโครงสร้างสถาปัตยกรรมรูปแบบเดิมที่ใช้กับเครื่องเล่นเกมส์ ให้มีความสามารถรองรับและสนับสนุนการเล่นเกมส์ผ่านระบบออนไลน์ได้ โดยภาพรวมสถาปัตยกรรมซอฟต์แวร์รูปแบบที่ใช้กับเครื่องเล่นเกมส์ ก่อนการเปลี่ยนแปลง สามารถแสดงได้ดังรูปที่ 5.5



รูปที่ 5.5 กรณีทดสอบที่ 3
ภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ก่อนการเปลี่ยนแปลง

เมื่อกลุ่มที่ปรึกษาได้ทำการเก็บข้อมูลต่างๆ เกี่ยวกับส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์เกมส์ในรูปแบบเดิม โดยนำข้อมูลส่วนประกอบทั้งหมดมาวิเคราะห์และออกแบบสถาปัตยกรรมซอฟต์แวร์เกมส์ ที่นำระบบการเล่นแบบออนไลน์มาใช้ กลุ่มที่ปรึกษาจึงมีข้อสรุปว่าบริษัทผู้ผลิตเกมส์ยังสามารถที่จะนำส่วนประกอบหลักของสถาปัตยกรรมซอฟต์แวร์เดิมกลับมาใช้งาน สำหรับโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่จะปรับเปลี่ยนต่อไปได้ โดยมีส่วนประกอบที่สามารถนำกลับมาใช้งานใหม่ อาทิเช่น ส่วนประกอบที่เกี่ยวข้องกับการแสดงผลของเกมส์, ส่วนประกอบที่เกี่ยวกับการบังคับการเล่นของเกมส์ และส่วนประกอบหลักที่ใช้ในการประมวลผลเกมส์ เป็นต้น โดยทางกลุ่มที่ปรึกษาจะมีการออกแบบส่วนประกอบที่จำเป็นต่อการจัดทำระบบเกมส์ออนไลน์เพิ่มเติม ได้แก่ ส่วนการติดต่อประสานงานกับส่วนประมวลผลเกมส์, ส่วนประกอบที่เกี่ยวข้องกับการทำงานของฐานข้อมูลหลัก และส่วนประกอบที่เกี่ยวข้องกับการทำงานของฐานข้อมูลลูกข่าย โดยภาพรวมสถาปัตยกรรมซอฟต์แวร์รูปแบบเกมส์ออนไลน์แบบใหม่ หลังการเปลี่ยนแปลง สามารถแสดงได้ดังรูปที่ 5.6



รูปที่ 5.6 กรณีทดสอบที่ 3
ภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ หลังการเปลี่ยนแปลง

ทั้งนี้ ข้อมูลที่กลุ่มที่ปรึกษาได้รวบรวมและวิเคราะห์ความต้องการที่เกี่ยวข้องกับการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ ได้รับการบันทึกข้อมูลโดยใช้เครื่องมือที่พัฒนาขึ้นสำหรับใช้เป็นข้อมูลเบื้องต้นสำหรับโครงการ

โดยตัวอย่างข้อมูลและผลการทดสอบเครื่องมือส่วนการจัดการคำอธิบายข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ สามารถแสดงตัวอย่างได้ ดังตารางที่ 5.21 - 5.24

ตารางที่ 5.21 ตัวอย่างและผลลัพธ์การบันทึกคำอธิบายโครงการ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ รหัสโครงการ ARCADE-CODE-A01

ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์				
คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์โครงการ	String	PK	สำเร็จ
code	รหัสโครงการ	String	ARCADE-CODE-A01	สำเร็จ
name	ชื่อโครงการ	String	ARCADE-GAME	สำเร็จ

ตารางที่ 5.21 ตัวอย่างและผลลัพธ์การบันทึกคำอธิบายโครงการ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ รหัสโครงการ ARCADE-CODE-A01 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
description	คำอธิบายโครงการ	String	Develops sample arcade architecture for study game engine activities	สำเร็จ
requirementID	ค่าคีย์ของคลาส Requirement	List	-	-
fileName	ชื่อแฟ้มข้อมูลโครงการ	String	-	-

ตารางที่ 5.22 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ที่ 1

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์ความต้องการ	String	1	สำเร็จ
code	รหัสความต้องการ	String	ARG-REQ01	สำเร็จ
name	ชื่อความต้องการ	String	Game engine development	สำเร็จ
description	คำอธิบายความต้องการ	String	-	สำเร็จ
status	สถานะความต้องการ	String	open	สำเร็จ
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U	สำเร็จ

ตารางที่ 5.22 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ที่ 1 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U	สำเร็จ
submittedDate	วันที่สร้าง	Date	28/12/2012	สำเร็จ
updatedDate	วันที่ปรับปรุงข้อมูล	Date	18/01/2013	สำเร็จ

ตารางที่ 5.23 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ที่ 2

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์ความต้องการ	String	2	สำเร็จ
code	รหัสความต้องการ	String	ARG-REQ02	สำเร็จ
name	ชื่อความต้องการ	String	Connector Input Testing	สำเร็จ
description	คำอธิบายความต้องการ	String	-	สำเร็จ
status	สถานะความต้องการ	String	open	สำเร็จ
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U	สำเร็จ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U	สำเร็จ
submittedDate	วันที่สร้าง	Date	28/12/2012	สำเร็จ
updatedDate	วันที่ปรับปรุงข้อมูล	Date	18/01/2013	สำเร็จ

ตารางที่ 5.24 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ที่ 3

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์ความต้องการ	String	3	สำเร็จ
code	รหัสความต้องการ	String	ARG-REQ03	สำเร็จ
name	ชื่อความต้องการ	String	Connector Display Testing	สำเร็จ
description	คำอธิบายความต้องการ	String	-	สำเร็จ
status	สถานะความต้องการ	String	open	สำเร็จ
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U	สำเร็จ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U	สำเร็จ
submittedDate	วันที่สร้าง	Date	28/12/2012	สำเร็จ
updatedDate	วันที่ปรับปรุงข้อมูล	Date	18/01/2013	สำเร็จ

ทั้งนี้ รายละเอียดข้อมูลการทดสอบในส่วนการบันทึกข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ทั้งหมดได้แสดงไว้ในภาคผนวก ก

เมื่อข้อมูลรายการข้อกำหนดความต้องการและข้อมูลเบื้องต้นของโครงการ ซึ่งได้รับจากผู้ที่เกี่ยวข้องกับโครงการได้ถูกกำหนดโดยกลุ่มที่ปรึกษาเรียบร้อยแล้ว ทีมงานนักออกแบบจึงดำเนินการแก้ไขและปรับปรุงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ โดยใช้ข้อมูลดังกล่าวเป็นพื้นฐาน อีกทั้ง ยังได้มีการใช้เครื่องมือที่พัฒนาขึ้นในการบันทึกคำอธิบายการออกแบบให้กับส่วนประกอบต่างๆ ของโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่ปรับปรุงแก้ไขเพิ่มเติมอีกด้วย ตัวอย่างข้อมูลและผลการทดสอบเครื่องมือส่วนการระบุคำอธิบายการออกแบบให้กับส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ สามารถแสดงตัวอย่างได้ ดังตารางที่ 5.25

ตารางที่ 5.25 ตัวอย่างและผลการบันทึกคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ รหัสโครงการ ARCADE-CODE-A01

ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์				
จาก สถาปัตยกรรมซอฟต์แวร์ : AG_v1				
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : AG_v2				
ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ	ผลการบันทึกข้อมูล
1	Game	1 - Game engine development	Functionality	สำเร็จ
2	KeyBoardDriver	2 - Connector Input Testing, 4 - Key input testing	Efficiency, Functionality, Reliability, Usability	สำเร็จ
3	MouseDriver	2 - Connector Input Testing, 4 - Key input testing	Efficiency, Functionality, Reliability, Usability	สำเร็จ
4	DisplayDriver	3 - Connector Display Testing 5 - Display output testing	Efficiency, Functionality, Reliability, Usability	สำเร็จ
5	ConntrollerEngine	6 - Online Game Implementation	Efficiency, Functionality, Maintainability, Portability	สำเร็จ

ทั้งนี้ รายละเอียดข้อมูลการทดสอบในส่วนการบันทึกคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์โครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ทั้งหมดได้แสดงไว้ในภาคผนวก ข

หลังจากทีมงานนักออกแบบสถาปัตยกรรมซอฟต์แวร์ได้ทำการปรับปรุงแก้ไขโครงสร้างสถาปัตยกรรมซอฟต์แวร์ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์แล้วเสร็จ บริษัทผู้ผลิต รวมถึงผู้ที่เกี่ยวข้องกับโครงการมีความต้องการทราบถึงผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่เกี่ยวข้องกับสถาปัตยกรรมซอฟต์แวร์เกมส์รูปแบบเดิม อีกทั้งชุดคำสั่งที่มีการพัฒนาสถาปัตยกรรมรูปแบบเดิมว่ามีผลกระทบในส่วนใดบ้าง นักออกแบบจึงได้ใช้เครื่องมือที่พัฒนาขึ้นในการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ เพื่อนำข้อมูลที่ได้จากการประมวลผลของเครื่องมือไปใช้รายงานกับผู้ที่เกี่ยวข้องกับโครงการต่อไป

โดยตัวอย่างผลลัพธ์การทดสอบเครื่องมือส่วนการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ สามารถแสดงตัวอย่างได้ ดังตารางที่ 5.26 - 5.30

ตารางที่ 5.26 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ส่วนผลกระทบต่อความต้องการซอฟต์แวร์

ผลกระทบต่อความต้องการซอฟต์แวร์		
จาก สถาปัตยกรรมซอฟต์แวร์ : AG_v1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : AG_v2		
ลำดับที่	ชื่อความต้องการ	ส่วนประกอบที่เกี่ยวข้อง
1	1 - Game engine development	Game
2	2 - Connector Input Testing	KeyBoardDriver, MouseDriver
3	3 - Connector Display Testing	DisplayDriver
4	4 - Key input testing	KeyBoardDriver, MouseDriver
5	5 - Display output testing	DisplayDriver

จากตัวอย่างผลลัพธ์การตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ ในส่วนผลกระทบต่อความต้องการซอฟต์แวร์ สามารถอธิบายได้ว่า มีรายการข้อกำหนดความต้องการที่ได้รับผลกระทบ คือ รายการข้อกำหนดความต้องการที่ 1 - Game engine development ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ Game และรายการข้อกำหนดความต้องการที่ 2 - Connector Input Testing ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ KeyBoardDriver และ MouseDriver เป็นต้น

ตารางที่ 5.27 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ

ผลกระทบต่อคุณลักษณะเชิงคุณภาพ		
จาก สถาปัตยกรรมซอฟต์แวร์ : AG_v1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : AG_v2		
ลำดับที่	ชื่อคุณลักษณะเชิงคุณภาพ	ส่วนประกอบที่เกี่ยวข้อง
1	Functionality	ConntrollerEngine, DatabaseClient, DisplayDriver, Game, KeyBoardDriver, MouseDriver, DatabaseServer
2	Maintainability	ConntrollerEngine, DatabaseClient, DatabaseServer
3	Portability	ConntrollerEngine, DatabaseClient, DatabaseServer
4	Reliability	DisplayDriver, KeyBoardDriver, MouseDriver
5	Usability	DisplayDriver, KeyBoardDriver, MouseDriver

จากตัวอย่างผลลัพธ์การตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ ในส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ สามารถอธิบายได้ว่า มีคุณลักษณะเชิงคุณภาพที่ได้รับผลกระทบ คือ คุณลักษณะเชิงคุณภาพด้านหน้าที่การทำงาน ด้านการบำรุงรักษา ด้านความสะดวกในการพกพา ด้านความน่าเชื่อถือ และด้านการใช้งาน ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ ConntrollerEngine, DatabaseClient, DisplayDriver, Game, KeyBoardDriver, MouseDriver และ DatabaseServer ตามลำดับ

ตารางที่ 5.28 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ส่วนผลกระทบต่อชุดคำสั่ง

ผลกระทบต่อชุดคำสั่ง		
จาก สถาปัตยกรรมซอฟต์แวร์ : AG_v1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : AG_v2		
ลำดับที่	ชื่อชุดคำสั่ง	ส่วนประกอบที่เกี่ยวข้อง
1	com.arcade.core.Engine	ConntrollerEngine, Game
2	com.arcade.Input	KeyBoardDriver, MouseDriver
3	com.arcade.Output	DisplayDriver

จากตัวอย่างผลลัพธ์การตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ ในส่วนผลกระทบต่อชุดคำสั่ง สามารถอธิบายได้ว่า มีชุดคำสั่งที่ได้รับผลกระทบ คือ com.arcade.core.Engine ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ ConntrollerEngine, Game และชุดคำสั่ง com.arcade.Input ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ KeyBoardDriver, MouseDriver และชุดคำสั่ง com.arcade.Output ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ DisplayDriver ตามลำดับ

ตารางที่ 5.29 ตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์

ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์			
จาก สถาปัตยกรรมซอฟต์แวร์ : AG_v1			
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : AG_v2			
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	ประเภทความเปลี่ยนแปลง
1	ConntrollerEngine	Component	Add-In
2	DatabaseClient	Component	Add-In
3	DatabaseServer	Component	Add-In
4	ConnectorCE	Connector	Add-In
5	ConnectorDBClient	Connector	Add-In

จากตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ สามารถอธิบายได้ว่า มีส่วนประกอบที่มีการออกแบบเพิ่มเติมขึ้นมา จากสถาปัตยกรรมตั้งต้น ได้แก่ ส่วนประกอบ ConntrollerEngine, DatabaseClient, DatabaseServer และมีการออกแบบตัวเชื่อมเพิ่มเติม คือ ตัวเชื่อม ConnectorCE และ ConnectorDBClient เป็นต้น โดยรายละเอียดผลการทดสอบทั้งหมดได้แสดงไว้ในภาคผนวก ค

ตารางที่ 5.30 ตัวอย่างการตรวจจับรายละเอียดการเปลี่ยนแปลงส่วนประกอบ ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์

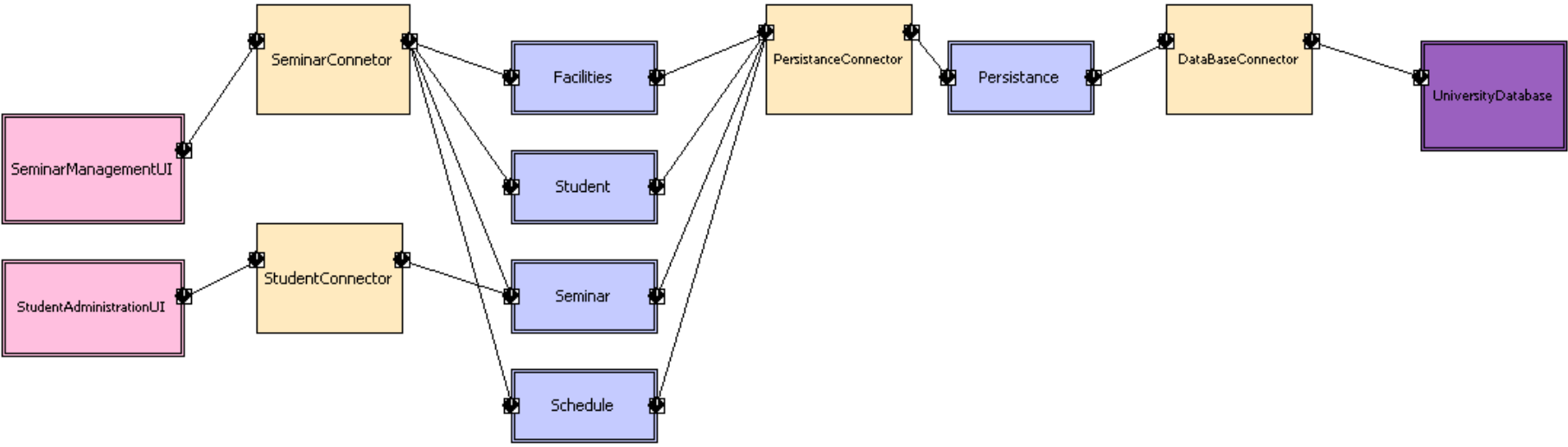
ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์					
จาก สถาปัตยกรรมซอฟต์แวร์ : AG_v1					
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : AG_v2					
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	การเปลี่ยนแปลงคุณสมบัติ	ค่าเดิม	ค่าใหม่
1	USBInf1	Interface	Description	USBInfInf1	USBInf1

จากตัวอย่างผลลัพธ์การตรวจจับรายละเอียดการเปลี่ยนแปลงส่วนประกอบ ที่ได้จากการทำงานของเครื่องมือ สามารถอธิบายได้ว่า มีส่วนประกอบที่มีการแก้ไขรายละเอียดจากสถาปัตยกรรมตั้งต้น ได้แก่ จุดเชื่อมต่อ USBInf1 โดยมีการแก้ไขคำอธิบาย จากเดิม คือ USBInfInf1 เปลี่ยนเป็น USBInf1

5.1.4 **กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย**

มหาวิทยาลัยเอกชนแห่งหนึ่ง มีความต้องการในการปรับปรุงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ระบบลงทะเบียนเดิมของมหาวิทยาลัยที่ผ่านการใช้งานมาเป็นระยะเวลานาน ซึ่งเกิดปัญหาเกี่ยวกับประสิทธิภาพในการทำงาน เนื่องจากการเชื่อมต่อของหน่วยงานต่างๆ เป็นไปในลักษณะการจับกลุ่มตามแต่ละหน่วยงานที่แยกออกจากส่วนงานอื่นๆ ไม่มีการออกแบบการทำงานให้สามารถเชื่อมต่อระหว่างหน่วยงานย่อยๆ ยกตัวอย่างเช่น ส่วนงานการจัดการข้อมูลการประชุมของมหาวิทยาลัย และส่วนงานการจัดการข้อมูลการประชุมของนักศึกษา ที่ทำงานแยกออกจากกัน อีกทั้ง ระบบยังไม่มีส่วนการทำงานในการบันทึกข้อมูลการใช้งานที่เกิดขึ้นกับระบบ ทำให้ผู้ดูแลระบบไม่สามารถตรวจสอบการทำงานของเจ้าหน้าที่ย้อนหลังได้ เป็นต้น จากปัญหาที่พบดังกล่าว หัวหน้าฝ่ายจัดการเทคโนโลยีสารสนเทศของมหาวิทยาลัยมีความพยายามในการแก้ไขปัญหาที่

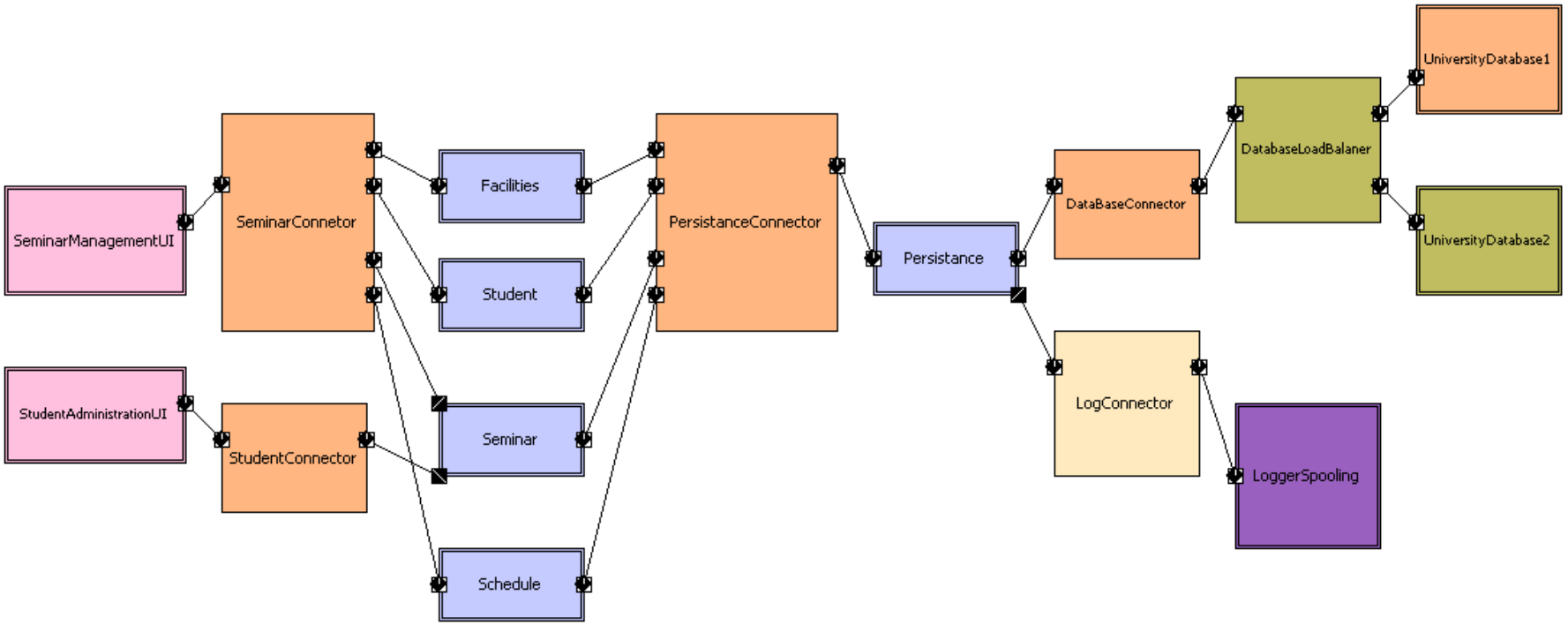
เกิดขึ้น เพื่อให้เจ้าหน้าที่ผู้ปฏิบัติงานสามารถใช้ระบบที่พัฒนาขึ้นเป็นเครื่องมือในการปฏิบัติงานให้เกิดประสิทธิภาพกับองค์กรมากยิ่งขึ้น นอกจากนี้ การแก้ไขปัญหาที่เกิดขึ้น ยังสามารถทำให้ผู้ดูแลระบบสามารถตรวจสอบการทำงานต่างๆ ระหว่างหน่วยงานภายในมหาวิทยาลัยย้อนหลังได้อีกด้วย ฝ่ายจัดการเทคโนโลยีสารสนเทศจึงได้จัดการประชุมระหว่างผู้ที่เกี่ยวข้องกับโครงการปรับปรุงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ของมหาวิทยาลัยขึ้น โดยภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบการลงทะเบียนมหาวิทยาลัย ก่อนการเปลี่ยนแปลง สามารถแสดงได้ดังรูปที่ 5.7



รูปที่ 5.7 กรณีทดสอบที่ 4

ภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบการลงทะเบียนมหาวิทยาลัย ก่อนการเปลี่ยนแปลง

หลังจากการประชุมดังกล่าวเสร็จสิ้นที่ประชุมจึงเห็นตรงกันว่า โครงสร้างสถาปัตยกรรมซอฟต์แวร์ใหม่ของมหาวิทยาลัยที่จะทำการปรับปรุงเพิ่มเติมนั้น จะต้องมีการแยกการเชื่อมต่อกันระหว่างการทำงานในลักษณะเดิม ให้มีความสอดคล้องต่อการปฏิบัติงานในปัจจุบัน และมีความยืดหยุ่นต่อการทำงาน อีกทั้งจะต้องเพิ่มเติม ส่วนประกอบในการบันทึกข้อมูลเป็นแฟ้มประวัติการดำเนินการของงานระเบียนและการจัดการประชุมต่างๆ ทั้งส่วนงานของมหาวิทยาลัยและส่วนงานของนักศึกษา นอกจากนี้ ยังต้องมีการปรับปรุงฐานข้อมูลให้มีความสมดุลมากยิ่งขึ้น เพื่อเพิ่มประสิทธิภาพการทำงานของระบบซอฟต์แวร์โดยรวม ทั้งนี้ เพื่อให้สถาปัตยกรรมซอฟต์แวร์ที่ออกแบบใหม่ มีความสามารถรองรับการทำงานและความต้องการขององค์กรที่อาจมีเพิ่มเติมมากขึ้นในอนาคต โดยภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบการลงทะเบียนมหาวิทยาลัย หลังการเปลี่ยนแปลง สามารถแสดงได้ดังรูปที่ 5.8



รูปที่ 5.8 กรณีทดสอบที่ 4

ภาพรวมสถาปัตยกรรมซอฟต์แวร์ระบบการลงทะเบียนมหาวิทยาลัย หลังการเปลี่ยนแปลง

ข้อมูลที่ฝ่ายเทคโนโลยีสารสนเทศ ได้รวบรวมและวิเคราะห์ความต้องการที่เกี่ยวข้องกับการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ดังกล่าว และได้ใช้เครื่องมือที่พัฒนาขึ้นในการบันทึกข้อมูลความต้องการทั้งหมด เพื่อใช้เป็นข้อมูลสำหรับโครงการ โดยตัวอย่างข้อมูลและผลการทดสอบเครื่องมือส่วนการจัดการคำอธิบายข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย สามารถแสดงตัวอย่างได้ ดังตารางที่ 5.31 - 5.34

ตารางที่ 5.31 ตัวอย่างและผลลัพธ์การบันทึกคำอธิบายโครงการ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย รหัสโครงการ UNI-CODE-2013

ระบบการลงทะเบียนมหาวิทยาลัย				
คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์โครงการ	String	PK	สำเร็จ
code	รหัสโครงการ	String	UNI-CODE-2013	สำเร็จ
name	ชื่อโครงการ	String	University management system 2013	สำเร็จ
description	คำอธิบายโครงการ	String	To develop university management system	สำเร็จ
requirementID	ค่าคีย์ของคลาส Requirement	List	-	-
fileName	ชื่อแฟ้มข้อมูลโครงการ	String	-	-

ตารางที่ 5.32 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ที่ 1

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์ความต้องการ	String	1	สำเร็จ
code	รหัสความต้องการ	String	UNI-REQ01	สำเร็จ
name	ชื่อความต้องการ	String	Development Seminar UI Component	สำเร็จ
description	คำอธิบายความต้องการ	String	-	สำเร็จ
status	สถานะความต้องการ	String	open	สำเร็จ
reporter	ชื่อผู้รายงานความต้องการ	String	Wiwat.V	สำเร็จ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U	สำเร็จ
submittedDate	วันที่สร้าง	Date	07/01/2013	สำเร็จ
updatedDate	วันที่ปรับปรุงข้อมูล	Date	14/02/2013	สำเร็จ

ตารางที่ 5.33 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ที่ 2

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์ความต้องการ	String	2	สำเร็จ
code	รหัสความต้องการ	String	UNI-REQ02	สำเร็จ
name	ชื่อความต้องการ	String	Development Student UI Component	สำเร็จ
description	คำอธิบายความต้องการ	String	-	สำเร็จ
status	สถานะความต้องการ	String	open	สำเร็จ

ตารางที่ 5.33 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ที่ 2 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
reporter	ชื่อผู้รายงานความต้องการ	String	Wiwat.V	สำเร็จ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U	สำเร็จ
submittedDate	วันที่สร้าง	Date	07/01/2013	สำเร็จ
updatedDate	วันที่ปรับปรุงข้อมูล	Date	14/02/2013	สำเร็จ

ตารางที่ 5.34 ตัวอย่างและผลลัพธ์การบันทึกข้อกำหนดความต้องการโครงการ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ที่ 3

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ	ผลการบันทึกข้อมูล
id	ค่าคีย์ความต้องการ	String	3	สำเร็จ
code	รหัสความต้องการ	String	UNI-REQ03	สำเร็จ
name	ชื่อความต้องการ	String	Development Facilities Entry Module	สำเร็จ
description	คำอธิบายความต้องการ	String	-	สำเร็จ
status	สถานะความต้องการ	String	open	สำเร็จ
reporter	ชื่อผู้รายงานความต้องการ	String	Wiwat.V	สำเร็จ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U	สำเร็จ
submittedDate	วันที่สร้าง	Date	07/01/2013	สำเร็จ
updatedDate	วันที่ปรับปรุงข้อมูล	Date	14/02/2013	สำเร็จ

ทั้งนี้ รายละเอียดข้อมูลการทดสอบในส่วนการบันทึกข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัยทั้งหมดได้แสดงไว้ในภาคผนวก ก

เมื่อข้อมูลความต้องการและข้อมูลต่างๆ จากผู้เกี่ยวข้องกับโครงการได้ถูกกำหนดเรียบร้อยแล้ว นักออกแบบสถาปัตยกรรมซอฟต์แวร์จึงดำเนินการแก้ไขและปรับปรุงโครงสร้างสถาปัตยกรรมซอฟต์แวร์โดยใช้ข้อมูลดังกล่าวเป็นพื้นฐาน อีกทั้งยังได้มีการใช้เครื่องมือที่พัฒนาขึ้นในการบันทึกคำอธิบายการออกแบบให้กับส่วนประกอบต่างๆ ของโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่ปรับปรุงแก้ไขเพิ่มเติมอีกด้วย ตัวอย่างข้อมูลและผลการทดสอบเครื่องมือส่วนการระบุคำอธิบายการออกแบบให้กับส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ระบบการลงทะเบียนมหาวิทยาลัย สามารถแสดงตัวอย่างได้ ดังตารางที่ 5.35

ตารางที่ 5.35 ตัวอย่างและผลการบันทึกคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย รหัสโครงการ UNI-CODE-2013

ระบบการลงทะเบียนมหาวิทยาลัย				
จาก สถาปัตยกรรมซอฟต์แวร์ : US_v1				
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : US_v2				
ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ	ผลการบันทึกข้อมูล
1	SeminarManagementUI	1 - Development Seminar UI Component	Functionality, Usability	สำเร็จ
2	StudentAdministrationUI	2 - Development Student UI Component	Functionality, Usability	สำเร็จ
3	Facilities	3 - Development Facilities Entry Module	Efficiency, Functionality, Maintainability, Reliability	สำเร็จ

ตารางที่ 5.35 ตัวอย่างและผลการบันทึกคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย รหัสโครงการ UNI-CODE-2013 (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ	ผลการบันทึกข้อมูล
4	Student	4 - Development Student Entry Module	Efficiency, Functionality, Maintainability, Reliability	สำเร็จ
5	Seminar	5 - Development Seminar Entry Module	Efficiency, Functionality, Maintainability, Reliability	สำเร็จ

ทั้งนี้ รายละเอียดข้อมูลการทดสอบในส่วนการบันทึกคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์โครงการระบบการลงทะเบียนมหาวิทยาลัยทั้งหมดได้แสดงไว้ในภาคผนวก ข

หลังจากทีมงานนักออกแบบสถาปัตยกรรมซอฟต์แวร์ได้ทำการปรับปรุงแก้ไขโครงสร้างสถาปัตยกรรมซอฟต์แวร์ระบบการลงทะเบียนมหาวิทยาลัยแล้วเสร็จ หัวหน้าฝ่ายเทคโนโลยีสารสนเทศ ทีมงานตรวจสอบคุณภาพ รวมถึงผู้ที่เกี่ยวข้องกับโครงการมีความต้องการทราบถึงผลกระทบจากการเปลี่ยนแปลงโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่เกี่ยวข้องกับสถาปัตยกรรมซอฟต์แวร์รูปแบบเดิมว่ามีผลกระทบในส่วนใดบ้าง นักออกแบบจึงได้ใช้เครื่องมือที่พัฒนาขึ้นในการตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ เพื่อนำข้อมูลที่ได้จากการประมวลผลของเครื่องมือไปใช้รายงานกับผู้ที่เกี่ยวข้องกับโครงการ รวมถึงจัดทำรายงานแสดงรายละเอียดการเปลี่ยนแปลงเพื่อใช้ในการตรวจสอบข้อมูลย้อนหลัง ต่อไป

ตัวอย่างผลลัพธ์การทดสอบเครื่องมือส่วนการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ระบบการลงทะเบียนมหาวิทยาลัย สามารถแสดงตัวอย่างได้ ดังตารางที่ 5.36 - 5.40

ตารางที่ 5.36 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ส่วนผลกระทบต่อความต้องการซอฟต์แวร์

ผลกระทบต่อความต้องการซอฟต์แวร์		
จาก สถาปัตยกรรมซอฟต์แวร์ : US_v1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : US_v2		
ลำดับที่	ชื่อความต้องการ	ส่วนประกอบที่เกี่ยวข้อง
1	1 - Development Seminar UI Component	SeminarManagementUI
2	2 - Development Student UI Component	StudentAdministrationUI
3	3 - Development Facilities Entry Module	Facilities
4	4 - Development Student Entry Module	Student
5	5 - Development Seminar Entry Module	Seminar

จากตัวอย่างผลลัพธ์การตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ ในส่วนผลกระทบต่อความต้องการซอฟต์แวร์ สามารถอธิบายได้ว่า มีรายการข้อกำหนดความต้องการที่ได้รับผลกระทบ คือ รายการข้อกำหนดความต้องการที่ 1 – Development Seminar UI Component ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ SeminarManagementUI และรายการข้อกำหนดความต้องการที่ 2 - Development Student UI Component ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ StudentAdministrationUI เป็นต้น

ตารางที่ 5.37 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ

ผลกระทบต่อคุณลักษณะเชิงคุณภาพ		
จาก สถาปัตยกรรมซอฟต์แวร์ : US_v1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : US_v2		
ลำดับที่	ชื่อคุณลักษณะเชิงคุณภาพ	ส่วนประกอบที่เกี่ยวข้อง
1	Efficiency	Facilities, Persistence, Seminar, Student, UniversityDatabase1, UniversityDatabase2, Schedule

ตารางที่ 5.37 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ (ต่อ)

ลำดับที่	ชื่อคุณลักษณะเชิงคุณภาพ	ส่วนประกอบที่เกี่ยวข้อง
2	Functionality	Facilities, LoggerSpooling, Persistence, Seminar, SeminarManagementUI, Student, StudentAdministrationUI, UniversityDatabase1, Schedule
3	Maintainability	Facilities, LoggerSpooling, Persistence, Seminar, Student, UniversityDatabase1, UniversityDatabase2, Schedule
4	Reliability	Facilities, LoggerSpooling, Persistence, Seminar, Student, UniversityDatabase1, UniversityDatabase2, Schedule
5	Usability	SeminarManagementUI, StudentAdministrationUI

จากตัวอย่างผลลัพธ์การตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ ในส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ สามารถอธิบายได้ว่า มีคุณลักษณะเชิงคุณภาพที่ได้รับผลกระทบ คือ คุณลักษณะเชิงคุณภาพด้านประสิทธิภาพ ด้านหน้าที่การทำงาน ด้านการบำรุงรักษา ด้านความน่าเชื่อถือ และด้านการใช้งาน ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ Facilities, Persistence, Seminar, Student, UniversityDatabase1, UniversityDatabase2, Schedule, LoggerSpooling, SeminarManagementUI และ StudentAdministrationUI ตามลำดับ

ตารางที่ 5.38 ตัวอย่างผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ส่วนผลกระทบต่อชุดคำสั่ง

ผลกระทบต่อชุดคำสั่ง		
จาก สถาปัตยกรรมซอฟต์แวร์ : US_v1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : US_v2		
ลำดับที่	ชื่อชุดคำสั่ง	ส่วนประกอบที่เกี่ยวข้อง
1	com.university.entry.businessProcess	Facilities, Seminar, Student, Schedule
2	com.university.entry.common.Utils	Persistence, UniversityDatabase1, UniversityDatabase2, LoggerSpooling
3	com.university.entry.UIService	SeminarManagementUI, StudentAdministrationUI

จากตัวอย่างผลลัพธ์การตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ ในส่วนผลกระทบต่อชุดคำสั่ง สามารถอธิบายได้ว่า มีชุดคำสั่งที่ได้รับผลกระทบคือ com.university.entry.businessProcess ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ Facilities, Seminar, Student, Schedule ชุดคำสั่ง com.university.entry.common.Utils ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ Persistence, UniversityDatabase1, UniversityDatabase2, LoggerSpooling และชุดคำสั่ง com.university.entry.UIService ซึ่งมีความเกี่ยวข้องกับส่วนประกอบ SeminarManagementUI, StudentAdministrationUI ตามลำดับ

ตารางที่ 5.39 ตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย

ระบบการลงทะเบียนมหาวิทยาลัย			
จาก สถาปัตยกรรมซอฟต์แวร์ : US_v1			
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : US_v2			
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	ประเภทความเปลี่ยนแปลง
1	LoggerSpooling	Component	Add-In

ตารางที่ 5.39 ตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	ประเภทความเปลี่ยนแปลง
2	UniversityDatabase2	Component	Add-In
3	DatabaseLoadBalancer	Connector	Add-In
4	LogConnector	Connector	Add-In
5	iDBL1	Interface	Add-In

จากตัวอย่างผลลัพธ์การตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่ได้จากการทำงานของเครื่องมือ สามารถอธิบายได้ว่า มีส่วนประกอบที่มีการออกแบบเพิ่มเติมขึ้นมา จากสถาปัตยกรรมตั้งต้น ได้แก่ ส่วนประกอบ LoggerSpooling, UniversityDatabase2 และมีการออกแบบตัวเชื่อมเพิ่มเติม คือ ตัวเชื่อม DatabaseLoadBalancer, LogConnector เป็นต้น

ตารางที่ 5.40 ตัวอย่างการตรวจจับรายละเอียดการเปลี่ยนแปลงส่วนประกอบ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย

ระบบการลงทะเบียนมหาวิทยาลัย					
จาก สถาปัตยกรรมซอฟต์แวร์ : US_v1					
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : US_v2					
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	การเปลี่ยนแปลงคุณสมบัติ	ค่าเดิม	ค่าใหม่
1	Persistence	Component	No of Interface	2	3
2	Seminar	Component	No of Interface	2	3
3	University Database1	Component	Description	University Database	University Database1
4	Persistence Connector	Connector	No of Interface	2	5

ตารางที่ 5.40 ตัวอย่างการตรวจจับรายละเอียดการเปลี่ยนแปลงส่วนประกอบ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	การเปลี่ยนแปลงคุณสมบัติ	ค่าเดิม	ค่าใหม่
5	SeminarConnetor	Connector	No of Interface	2	5

จากตัวอย่างผลลัพธ์การตรวจจับรายละเอียดการเปลี่ยนแปลงส่วนประกอบ ที่ได้จากการทำงานของเครื่องมือ สามารถอธิบายได้ว่า มีส่วนประกอบที่มีการแก้ไขรายละเอียดจากสถาปัตยกรรมตั้งต้น ได้แก่ ส่วนประกอบ Persistence โดยมีการแก้ไขจำนวนจุดเชื่อมต่อ จากเดิม คือ 2 จุด เปลี่ยนเป็น 3 จุด และส่วนประกอบ Seminar มีการแก้ไขจำนวนจุดเชื่อมต่อ จากเดิม คือ 2 จุด เปลี่ยนเป็น 3 จุด เป็นต้น โดยรายละเอียดผลการทดสอบทั้งหมดได้แสดงไว้ในภาคผนวก ค

บทที่ 6
สรุปผลการวิจัย

ในบทนี้กล่าวถึงผลสรุปการวิจัย ประโยชน์ที่ได้รับจากงานวิจัย ข้อจำกัดของเครื่องมือ และแนวทางในการพัฒนางานวิจัยต่อไปในอนาคต ซึ่งมีรายละเอียดดังต่อไปนี้

6.1 สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนอวิธีในการออกแบบสถาปัตยกรรมซอฟต์แวร์ที่ใช้เป็นทางเลือกสำหรับผลิตภัณฑ์ซอฟต์แวร์ที่มีขนาดใหญ่ จากการเพิ่มเติมกิจกรรมที่มีความสำคัญรวมเข้ากับแบบจำลองการออกสถาปัตยกรรมซอฟต์แวร์รูปแบบเดิม กิจกรรมที่มีการเพิ่มเติมดังกล่าว ได้รับการสนับสนุนการทำงานโดยเครื่องมือที่พัฒนาขึ้น ซึ่งมีชื่อว่า “xADL: Software Architecture Changes Effect Detection Tools – xSACEDT” โดยหลักการทำงานของเครื่องมือยังมีส่วนเกี่ยวข้องกับคำอธิบายข้อกำหนดความต้องการโครงการ คำอธิบายและคุณลักษณะเชิงคุณภาพที่เกี่ยวกับการออกแบบส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ที่ถูกกำหนดเป็นคำอธิบายเพิ่มเติมแนบไปกับแบบจำลองสถาปัตยกรรมซอฟต์แวร์ในภาษาเอ็กซ์เอดีแอล การทดสอบเครื่องมือที่ใช้ในการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กระทำโดยการกำหนดกรณีทดสอบ ทั้งหมด 4 กรณีทดสอบ ซึ่งเครื่องมือสามารถตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ของกรณีทดสอบทุกกรณีได้

6.2 ประโยชน์ที่ได้รับจากงานวิจัย

1. ได้เครื่องมือสำหรับตรวจจับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ที่เกี่ยวข้องกัน ช่วยให้ผู้ที่เกี่ยวข้องในโครงการพัฒนาซอฟต์แวร์นำข้อมูลไปใช้ในการวิเคราะห์ผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์

2. ได้เครื่องมือสำหรับตรวจจับความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ ช่วยให้การออกแบบสถาปัตยกรรมเป็นไปอย่างครบถ้วนตรงตามข้อกำหนดความต้องการ อีกทั้งช่วยลดข้อผิดพลาดของการตรวจสอบความเปลี่ยนแปลงด้วยบุคคลสำหรับสถาปัตยกรรมซอฟต์แวร์ที่มีความซับซ้อน

3. ได้ข้อมูลผลสรุปการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ในรูปแบบรายงานที่มีความชัดเจนถูกต้อง ซึ่งสามารถนำไปใช้เป็นข้อมูลเบื้องต้นในการอ้างอิงกับผู้ที่เกี่ยวกับโครงการพัฒนาซอฟต์แวร์

6.3 ข้อจำกัดของเครื่องมือ

1. แบบจำลองสถาปัตยกรรมซอฟต์แวร์ที่ใช้ในตรวจจับผลกระทบของการเปลี่ยนแปลง รองรับแฟ้มข้อมูลนำเข้าในภาษาเอ็กซ์เอดีแอลเฉพาะเวอร์ชัน 2.0 เท่านั้น

2. การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์แบบจำลองสถาปัตยกรรมซอฟต์แวร์ที่นำมาใช้ตรวจจับต้องได้รับการกำหนดคำอธิบายการออกแบบให้กับส่วนประกอบจากขั้นตอนการจัดการคำอธิบายการออกแบบ

3. การตรวจจับและสรุปผลความเปลี่ยนแปลง เป็นการตรวจจับความเปลี่ยนแปลงด้านโครงสร้างสถาปัตยกรรมซอฟต์แวร์ ผลสรุปดังกล่าว ไม่ครอบคลุมในส่วนของการวิเคราะห์ด้านคุณภาพ

6.4 แนวทางในการพัฒนางานวิจัยต่อ

1. พัฒนาให้เครื่องมือมีความสามารถในการตรวจจับผลการเปลี่ยนแปลงกับภาษาเอ็กซ์เอดีแอลในเวอร์ชัน 3.0

2. เพิ่มเติมตัวชี้วัดใดๆ เพื่อระบุค่าน้ำหนักสำหรับการเปลี่ยนแปลงส่วนประกอบสถาปัตยกรรมซอฟต์แวร์ และใช้ข้อมูลของค่าน้ำหนักที่กำหนดนำมาวิเคราะห์ผลเกี่ยวกับคุณภาพของการเปลี่ยนแปลงที่เกิดขึ้น

3. พัฒนาความสามารถของเครื่องมือเพิ่มเติม เพื่อปรับเปลี่ยนการแสดงผลแบบจำลองเป็นลักษณะรูปภาพ เช่น การเพิ่มเติมสีให้กับส่วนประกอบที่ได้รับผลกระทบจากการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ เป็นต้น

รายการอ้างอิง

[1] SEI, Software Engineering Institute. What Is Software Architecture [Online]. 2013. Available from : http://www.sei.cmu.edu/architecture/ [2013,May 3]

[2] Bosch, J.,and Molin, P. Software architecture design: evaluation and transformation. IEEE Conf. and Workshop on Engineering of Computer-Based Systems, 1999.

[3] Mei Rong, M., Liu, C.,and Zhang, G. Modeling Aspect-oriented Software Architecture Based on ACME. The 6th International Conference on Computer Science & Education (ICCSE 2011), 2011.

[4] SEI, Software Engineering Institute. Architecture Evolution [Online]. 2013. Available from : http://www.sei.cmu.edu/architecture/research/previousresearch/evolution.cfm [2012,May 3]

[5] ISR, Institute for Software Research. Architecture Description Languages (ADL's) [Online]. 1997. Available from : http://www.isr.uci.edu/architecture/adls.html [2012,July 1]

[6] Dashofy, E. M.,and Van der Hoek, A. A highly-extensible, XML-based architecture description language. IEEE/IFIP Conference on Software Architecture, 2001.

[7] Bray, T., Paoli Sperberg-McQueen, J.,and Maler E. Extensible Markup Language (XML) 1.0 (Second Edition) [Online]. 2000. Available from : http://www.w3.org/TR/2000/REC-xml-20001006 [2012,Oct 6]

[8] ISR, Institute for Software Research. XML schema-to-Java data binding generator (Apigen) [Online]. 2000. Available from : http://www.isr.uci.edu/projects/xarchuci/tools-apigen.html [2012,June 24]

[9] ISR, Institute for Software Research. XML-based representation for software architectures (xArch) [Online]. 2000. Available from : http://www.isr.uci.edu/architecture/xarch/ [2012,Oct 26]

[10] ISR, Institute for Software Research. Archstudio 4.0 tool set for the xadl language [Online]. 2012. Available from : http://www.isr.uci.edu/projects/archstudio-4/www/archstudio/ [2012,Oct 1]

[11] ISR, Institute for Software Research. Archstudio 5.0 tool set for the xadl language. : [Online]. 2013. Available from : http://www.isr.uci.edu/projects/archstudio/ [2013,Feb 1]

[12] Guntersdorfer, M., Oreizy, P., Medvidovic N.,and Taylor R.N. xADL: enabling architecture-centric tool integration with XML. International Conference on System Sciences (HICSS'01), 2001.

[13] Fallside, D. C. XML Schema Part 0: Primer. [Online]. 2004. Available from : http://www.w3.org/TR/xmlschema-0/ [2012,Oct 28]

[14] W3C, World Wide Web Consortium. Namespaces in XML. [Online]. 2009. Available from : http://www.w3.org/TR/REC-xml-names/ [2012,Dec 8]

[15] International Organization for Standarization. ISO Standard 9126: Software Engineering – Product Quality, parts 1, 2 and 3. , International Organization for Standarization, Geneve, 2001.

[16] Delinchant, B.,and Gerbaud, L. Concurrent design versioning system, based on XML file. Annual Conference of the IEEE Industrial Electronics Society (IECON'02), 2002.

[17] Morris J. Hash Table. [Online]. 1998. Available from : http://www.cs.auckland.ac.nz/software/AlgAnim/hash_tables.html [2012,Oct 1]

[18] Boucke, N., Garcia, A.,and Holvoet T. Composing architectural crosscutting structures in xADL. Early Aspects at ICSE 2007: Workshop on Aspect-Oriented Requirements Engineering and Architecture Design (ICSE'07), 2007.

[19] Spencer, J. Architecture Description Markup Language (ADML). [Online]. 2000. Available from : http://www.opengroup.org/public/arch/p4/adml/adml_intro.htm [2012,Oct 1]

[20] W3C, World Wide Web Consortium. Document Object Model (DOM). [Online]. 2001. Available from : http://www.w3.org/DOM/ 2001 [2012,July 8]

[21] Megginson, D. Simple API for XML (SAX). [Online]. 2001. Available from : http://www.saxproject.org/ [2012,July 1]

[22] Hunter, J.,and McLaughlin B. Java-based document object model for XML (JDOM). [Online]. 2001. Available from : http://www.jdom.org/index.html., 2000. [2012,July 8]

[23] Van der Hoek, A. Capturing Product Line Architectures. In Proceedings of the 4 th International Software Architecture Workshop (ISAW-4'07), 2000.

[24] Antony T., Yan J.,and Jun H. A rationale-based architecture model for design traceability and reasoning. Journal of Systems and Software (JSS), Volume 80, (January 2007) : 918-934

[25] Dominic, J. Comma-separated values (CSV). [Online]. 2002. Available from : http://www.creativyst.com/Doc/Articles/CSV/CSV01.htm [2012,Sep 1]

[26] Garlan, D., Barnes, J., Schmerl, B.,and Celiku, O. Evolution Styles: Foundations and Tool Support for Software Architecture Evolution. European Conference on Software Architecture (ECSA'09), 2009.

[27] SEI, Software Engineering Institute. Arcade Game Maker Pedagogical Product Line: Architecture Documentation, Volume 2 - Software Architecture Views. [Online]. 2009. Available from : http://www.sei.cmu.edu/productlines/ppl/software_architecture_views.html/ [2013,Feb 12]

[28] Ambler, S., Lines, Mark. Disciplined Agile Delivery (DAD): A Practitioner's Guide to Agile Software Delivery in the Enterprise. : IBM Press., 2012.

ภาคผนวก

ภาคผนวก ก
คำอธิบายโครงการและคำอธิบายข้อกำหนดความต้องการโครงการ ระบบที่ใช้ในการทดสอบ

รายละเอียดคำอธิบายโครงการและคำอธิบายข้อกำหนดความต้องการโครงการของระบบที่ในการทดสอบ ใช้ข้อมูลในการทดสอบดังต่อไปนี้

ตารางที่ ก-1 คำอธิบายโครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M01

ระบบบริหารงานบริษัทเอบีซี			
คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์โครงการ	String	PK
code	รหัสโครงการ	String	ABC-CODE-M01
name	ชื่อโครงการ	String	ABC-M1
description	คำอธิบายโครงการ	String	Develops software solutions for continuous monitoring of ABC?s business activities
requirementID	ค่าคีย์ของคลาส Requirement	List	-
fileName	ชื่อแฟ้มข้อมูลโครงการ	String	-

ตารางที่ ก-2 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 1

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	1
code	รหัสความต้องการ	String	ABC-REQ01
name	ชื่อความต้องการ	String	Use for company account processing balance
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-3 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 2

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	2
code	รหัสความต้องการ	String	ABC-REQ02

ตารางที่ ก-3 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 2 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
name	ชื่อความต้องการ	String	Use for payback expense and automatic accounting balance
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	resolved
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-4 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 3

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	3
code	รหัสความต้องการ	String	ABC-REQ03
name	ชื่อความต้องการ	String	Use for salary payment and automatic accounting balance

ตารางที่ ก-4 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 3 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	resolved
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-5 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 4

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	4
code	รหัสความต้องการ	String	ABC-REQ04
name	ชื่อความต้องการ	String	Use for supplier payment by export
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	closed
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-6 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 5

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	5
code	รหัสความต้องการ	String	ABC-REQ05
name	ชื่อความต้องการ	String	Use for print customer payment slip
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	closed
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-7 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 6

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	6
code	รหัสความต้องการ	String	ABC-REQ06
name	ชื่อความต้องการ	String	Use for inventory automatic balance
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	resolved
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M

ตารางที่ ก-7 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 6 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-8 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 7

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	7
code	รหัสความต้องการ	String	ABC-REQ07
name	ชื่อความต้องการ	String	Use for office stuff and automatic update current of office stock
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-9 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 8

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	8
code	รหัสความต้องการ	String	ABC-REQ08
name	ชื่อความต้องการ	String	Use for enquiry employee historical data
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-10 คำอธิบายโครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M02

ระบบบริหารงานบริษัทเอบีซี			
คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์โครงการ	String	PK
code	รหัสโครงการ	String	ABC-CODE-M02
name	ชื่อโครงการ	String	ABC-M2

ตารางที่ ก-10 คำอธิบายโครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M02 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
description	คำอธิบายโครงการ	String	Maintain software architecture for ABC?s adaptive business activities
requirementID	ค่าคีย์ของคลาส Requirement	List	-
fileName	ชื่อแฟ้มข้อมูลโครงการ	String	-

ตารางที่ ก-11 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 1

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	1
code	รหัสความต้องการ	String	ABC-REQ01
name	ชื่อความต้องการ	String	Use for company account processing balance
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-12 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 2

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	2
code	รหัสความต้องการ	String	ABC-REQ02
name	ชื่อความต้องการ	String	Use for payback expense and automatic accounting balance
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	resolved
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-13 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 3

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	3
code	รหัสความต้องการ	String	ABC-REQ03

ตารางที่ ก-13 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 3 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
name	ชื่อความต้องการ	String	Use for salary payment and automatic accounting balance
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	resolved
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-14 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 4

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	4
code	รหัสความต้องการ	String	ABC-REQ04
name	ชื่อความต้องการ	String	Use for supplier payment by export
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	closed
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M

ตารางที่ ก-14 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 4 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-15 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 5

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	5
code	รหัสความต้องการ	String	ABC-REQ05
name	ชื่อความต้องการ	String	Use for print customer payment slip
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	closed
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-16 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 6

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	6
code	รหัสความต้องการ	String	ABC-REQ06

ตารางที่ ก-16 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 6 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
name	ชื่อความต้องการ	String	Use for inventory automatic balance
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	resolved
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-17 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 7

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	7
code	รหัสความต้องการ	String	ABC-REQ07
name	ชื่อความต้องการ	String	Use for office stuff and automatic update current of office stock
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M

ตารางที่ ก-17 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 7 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-18 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 8

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	8
code	รหัสความต้องการ	String	ABC-REQ08
name	ชื่อความต้องการ	String	Use for enquiry employee historical data
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Saran.M
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Saran.M
submittedDate	วันที่สร้าง	Date	21/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	21/12/2012

ตารางที่ ก-19 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 9

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	9
code	รหัสความต้องการ	String	ABC-REQ09
name	ชื่อความต้องการ	String	Use for payment business workflow grouping
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	13/02/2013
updatedDate	วันที่ปรับปรุงข้อมูล	Date	13/02/2013

ตารางที่ ก-20 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 10

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	10
code	รหัสความต้องการ	String	ABC-REQ10
name	ชื่อความต้องการ	String	Use for personnel business workflow grouping
description	คำอธิบายความต้องการ	String	-

ตารางที่ ก-20 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 10 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	13/02/2013
updatedDate	วันที่ปรับปรุงข้อมูล	Date	13/02/2013

ตารางที่ ก-21 ข้อกำหนดความต้องการโครงการระบบบริหารงานบริษัทเอบีซี ที่ 11

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	11
code	รหัสความต้องการ	String	ABC-REQ11
name	ชื่อความต้องการ	String	Use for inventory business workflow grouping
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	13/02/2013
updatedDate	วันที่ปรับปรุงข้อมูล	Date	13/02/2013

ตารางที่ ก-22 คำอธิบายโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์

ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์			
คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์โครงการ	String	PK
code	รหัสโครงการ	String	ARCADE-CODE-A01
name	ชื่อโครงการ	String	ARCADE-GAME
description	คำอธิบายโครงการ	String	Develops sample arcade architecture for study game engine activities
requirementID	ค่าคีย์ของคลาส Requirement	List	-
fileName	ชื่อแฟ้มข้อมูลโครงการ	String	-

ตารางที่ ก-23 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ที่ 1

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	1
code	รหัสความต้องการ	String	ARG-REQ01
name	ชื่อความต้องการ	String	Game engine development

ตารางที่ ก-23 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ที่ 1 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	28/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	18/01/2013

ตารางที่ ก-24 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ที่ 2

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	2
code	รหัสความต้องการ	String	ARG-REQ02
name	ชื่อความต้องการ	String	Connector Input Testing
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	28/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	18/01/2013

ตารางที่ ก-25 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ที่ 3

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	3
code	รหัสความต้องการ	String	ARG-REQ03
name	ชื่อความต้องการ	String	Connector Display Testing
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	28/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	18/01/2013

ตารางที่ ก-26 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ที่ 4

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	4
code	รหัสความต้องการ	String	ARG-REQ04
name	ชื่อความต้องการ	String	Key input testing
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	28/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	18/01/2013

ตารางที่ ก-27 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ที่ 5

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	5
code	รหัสความต้องการ	String	ARG-REQ05
name	ชื่อความต้องการ	String	Display output testing
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	28/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	18/01/2013

ตารางที่ ก-28 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ที่ 6

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	6
code	รหัสความต้องการ	String	ARG-REQ06
name	ชื่อความต้องการ	String	Online Game Implementation
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Artit.U
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U

ตารางที่ ก-28 ข้อกำหนดความต้องการโครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ที่ 6 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
submittedDate	วันที่สร้าง	Date	28/12/2012
updatedDate	วันที่ปรับปรุงข้อมูล	Date	18/01/2013

ตารางที่ ก-29 คำอธิบายโครงการระบบการลงทะเบียนมหาวิทยาลัย

ระบบการลงทะเบียนมหาวิทยาลัย			
คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์โครงการ	String	PK
code	รหัสโครงการ	String	UNI-CODE-2013
name	ชื่อโครงการ	String	University management system 2013
description	คำอธิบายโครงการ	String	To develop university management system
requirementID	ค่าคีย์ของคลาส Requirement	List	-
fileName	ชื่อแฟ้มข้อมูลโครงการ	String	-

ตารางที่ ก-30 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย ที่ 1

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	1
code	รหัสความต้องการ	String	UNI-REQ01

ตารางที่ ก-30 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย ที่ 1 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
name	ชื่อความต้องการ	String	Development Seminar UI Component
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Wiwat.V
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	07/01/2013
updatedDate	วันที่ปรับปรุงข้อมูล	Date	14/02/2013

ตารางที่ ก-31 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย ที่ 2

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	2
code	รหัสความต้องการ	String	UNI-REQ02
name	ชื่อความต้องการ	String	Development Student UI Component
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Wiwat.V
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	07/01/2013
updatedDate	วันที่ปรับปรุงข้อมูล	Date	14/02/2013

ตารางที่ ก-32 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย ที่ 3

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	3
code	รหัสความต้องการ	String	UNI-REQ03
name	ชื่อความต้องการ	String	Developme nt Facilities Entry Module
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Wiwat.V
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	07/01/2013
updatedDate	วันที่ปรับปรุงข้อมูล	Date	14/02/2013

ตารางที่ ก-33 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย ที่ 4

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	4
code	รหัสความต้องการ	String	UNI-REQ04
name	ชื่อความต้องการ	String	Developme nt Student Entry Module
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Wiwat.V
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	07/01/2013
updatedDate	วันที่ปรับปรุงข้อมูล	Date	14/02/2013

ตารางที่ ก-34 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย ที่ 5

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	5
code	รหัสความต้องการ	String	UNI-REQ05
name	ชื่อความต้องการ	String	Developme nt Seminar Entry Module
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Wiwat.V
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	07/01/2013
updatedDate	วันที่ปรับปรุงข้อมูล	Date	14/02/2013

ตารางที่ ก-35 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย ที่ 6

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	6
code	รหัสความต้องการ	String	UNI-REQ06
name	ชื่อความต้องการ	String	Developme nt Schedule Entry Module
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Wiwat.V
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	07/01/2013
updatedDate	วันที่ปรับปรุงข้อมูล	Date	14/02/2013

ตารางที่ ก-36 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย ที่ 7

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	7
code	รหัสความต้องการ	String	UNI-REQ07
name	ชื่อความต้องการ	String	Developme nt Common Persistence Entry Module
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Wiwat.V
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U
submittedDate	วันที่สร้าง	Date	07/01/2013
updatedDate	วันที่ปรับปรุงข้อมูล	Date	14/02/2013

ตารางที่ ก-37 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย ที่ 8

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
id	ค่าคีย์ความต้องการ	String	8
code	รหัสความต้องการ	String	UNI-REQ08
name	ชื่อความต้องการ	String	Developme nt Personal Security Module
description	คำอธิบายความต้องการ	String	-
status	สถานะความต้องการ	String	open
reporter	ชื่อผู้รายงานความต้องการ	String	Wiwat.V
assignedTo	ชื่อผู้รับผิดชอบต่อความต้องการ	String	Artit.U

ตารางที่ ก-37 ข้อกำหนดความต้องการโครงการระบบการลงทะเบียนมหาวิทยาลัย ที่ 8 (ต่อ)

คุณลักษณะ	รายละเอียด	ชนิด	ค่าที่จัดเก็บ
submittedDate	วันที่สร้าง	Date	07/01/2013
updatedDate	วันที่ปรับปรุงข้อมูล	Date	14/02/2013

ภาคผนวก ข
คำอธิบายการออกแบบสถาปัตยกรรมซอฟต์แวร์ระบบที่ใช้ในการทดสอบ

รายละเอียดคำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์ จากระบบที่ในการทดสอบ ใช้ข้อมูลในการทดสอบดังต่อไปนี้

ตารางที่ ข-1 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 1 โครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M01

ระบบบริหารงานบริษัทเอบีซี			
ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ
1	Accounts	1 - Use for company account processing balance	Efficiency, Maintainability, Reliability, Portability
2	Reimburse_ Expense	2 - Use for payback expense and automatic accounting balance	
3	Pay_Salary	3 - Use for salary payment and automatic accounting balance	Functionality
4	Pay_Supplier	4 - Use for supplier payment by export	Reliability
5	Bill_Customers	5 - Use for print customer payment slip	Usability
6	Inventory	6 - Use for inventory automatic balance	Efficiency
7	Supply_Office	7 - Use for office stuff and automatic update current of office stock	Maintainability

ตารางที่ ข-1 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 1 โครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M01 (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ
8	Personnel	8 - Use for enquiry employee historical data	Portability

ตารางที่ ข-2 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 1 โครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M02

ระบบบริหารงานบริษัทเอบีซี			
ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ
1	Accounts	1 - Use for company account processing balance	Efficiency, Maintainability, Reliability, Portability
2	Benefit_Expense	2 - Use for payback expense and automatic accounting balance	
3	Pay_Salary	3 - Use for salary payment and automatic accounting balance	Functionality
4	Pay_Supplier	4 - Use for supplier payment by export	Efficiency, Functionality, Reliability
5	Bill_Customers	5 - Use for print customer payment slip	Usability
6	Inventory	6 - Use for inventory automatic balance	Efficiency

ตารางที่ ข-2 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 1 โครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M02 (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ
7	Supply_Office	7 - Use for office stuff and automatic update current of office stock	Maintainability
8	Personnel	8 - Use for enquiry employee historical data	Portability

ตารางที่ ข-3 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 2 โครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M01

ระบบบริหารงานบริษัทเอบีซี			
ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ
1	Accounts	1 - Use for company account processing balance	Efficiency, Maintainability, Reliability, Portability
2	Reimburse_ Expense	2 - Use for payback expense and automatic accounting balance	
3	Pay_Salary	3 - Use for salary payment and automatic accounting balance	Functionality
4	Pay_Supplier	4 - Use for supplier payment by export	Reliability

ตารางที่ ข-3 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 2 โครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M01 (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ
5	Bill_Customers	5 - Use for print customer payment slip	Usability
6	Inventory	6 - Use for inventory automatic balance	Efficiency
7	Supply_Office	7 - Use for office stuff and automatic update current of office stock	Maintainability
8	Personnel	8 - Use for enquiry employee historical data	Portability

ตารางที่ ข-4 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 2 โครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M02

ระบบบริหารงานบริษัทเอบีซี			
ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ
1	Accounts	1 - Use for company account processing balance	Efficiency, Maintainability, Reliability, Portability
2	Reimburse_ Expense	2 - Use for payback expense and automatic accounting balance	
3	Pay_Salary	3 - Use for salary payment and automatic accounting balance	Functionality

ตารางที่ ข-4 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 2 โครงการระบบบริหารงานบริษัทเอบีซี รหัสโครงการ ABC-CODE-M02 (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ
4	Pay_Supplier	4 - Use for supplier payment by export	Reliability
5	Bill_Customers	5 - Use for print customer payment slip	Usability
6	Inventory	6 - Use for inventory automatic balance	Efficiency
7	Supply_Office	7 - Use for office stuff and automatic update current of office stock	Maintainability
8	Personnel	8 - Use for enquiry employee historical data	Portability
9	PaymentInterface	9 - Use for payment business workflow grouping	Maintainability
10	PersonnelInterface	10 - Use for personnel business workflow grouping	Maintainability
11	InventoryInterface	11 - Use for inventory business workflow grouping	Maintainability

ตารางที่ ข-5 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์

กรณีทดสอบที่ 3 โครงการระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ รหัสโครงการ ARCADE-CODE-A01

ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์			
ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ
1	Game	1 - Game engine development	Functionality
2	KeyBoardDriver	2 - Connector Input Testing, 4 - Key input testing	Efficiency, Functionality, Reliability, Usability
3	MouseDriver	2 - Connector Input Testing, 4 - Key input testing	Efficiency, Functionality, Reliability, Usability
4	DisplayDriver	3 - Connector Display Testing 5 - Display output testing	Efficiency, Functionality, Reliability, Usability
5	ConntrollerEngine	6 - Online Game Implementation	Efficiency, Functionality, Maintainability, Portability
6	DatabaseClient	6 - Online Game Implementation	Efficiency, Functionality, Maintainability, Portability
7	DatabaseServer	6 - Online Game Implementation	Efficiency, Functionality, Maintainability, Portability

ตารางที่ ข-6 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 4 โครงการระบบการลงทะเบียนมหาวิทยาลัย รหัสโครงการ UNI-CODE-2013

ระบบการลงทะเบียนมหาวิทยาลัย			
ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ
1	SeminarManagementUI	1 - Development Seminar UI Component	Functionality, Usability
2	StudentAdministrationUI	2 - Development Student UI Component	Functionality, Usability
3	Facilities	3 - Development Facilities Entry Module	Efficiency, Functionality, Maintainability, Reliability
4	Student	4 - Development Student Entry Module	Efficiency, Functionality, Maintainability, Reliability
5	Seminar	5 - Development Seminar Entry Module	Efficiency, Functionality, Maintainability, Reliability
6	Schedule	6 - Development Schedule Entry Module	Efficiency, Functionality, Maintainability, Reliability
7	Persistence	7 - Development Common Persistence Entry Module	Efficiency, Functionality, Maintainability, Reliability

ตารางที่ ข-6 คำอธิบายการออกแบบส่วนประกอบของสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 4 โครงการระบบการลงทะเบียนมหาวิทยาลัย รหัสโครงการ UNI-CODE-2013 (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ชื่อความต้องการ	คุณลักษณะเชิงคุณภาพ
8	UniversityDatabase1	7 - Development Common Persistence Entry Module	Efficiency, Functionality, Maintainability, Reliability
9	UniversityDatabase2	8 - Development Personal Security Module	Efficiency, Maintainability, Reliability
10	LoggerSpooling	-	Functionality, Maintainability, Reliability

ภาคผนวก ค
ผลการทดสอบการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์

รายละเอียดผลการทดสอบการตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์จากระบบที่ในการทดสอบ สามารถแสดงผลการทดสอบแยกตามระบบที่ใช้เป็นข้อมูลในการทดสอบดังต่อไปนี้

ตารางที่ ค-1 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อความต้องการซอฟต์แวร์

ผลกระทบต่อความต้องการซอฟต์แวร์		
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2		
ลำดับที่	ชื่อความต้องการ	ส่วนประกอบที่เกี่ยวข้อง
1	1 - Use for company account processing balance	Accounts
2	2 - Use for payback expense and automatic accounting balance	Benefit_Expense
3	3 - Use for salary payment and automatic accounting balance	Pay_Salary
4	4 - Use for supplier payment by export	Pay_Supplier
5	5 - Use for print customer payment slip	Bill_Customers
6	6 - Use for inventory automatic balance	Inventory
7	7 - Use for office stuff and automatic update current of office stock	Supply_Office
8	8 - Use for enquiry employee historical data	Personnel

ตารางที่ ค-2 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ

ผลกระทบต่อคุณลักษณะเชิงคุณภาพ		
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2		
ลำดับที่	ชื่อคุณลักษณะเชิงคุณภาพ	ส่วนประกอบที่เกี่ยวข้อง
1	Efficiency	Accounts, Inventory, Pay_Supplier
2	Functionality	Pay_Salary, Pay_Supplier
3	Maintainability	Accounts, Supply_Office
4	Portability	Accounts, Personnel
5	Reliability	Accounts, Pay_Supplier
6	Usability	Bill_Customers

ตารางที่ ค-3 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อชุดคำสั่ง

ผลกระทบต่อชุดคำสั่ง		
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2		
ลำดับที่	ชื่อชุดคำสั่ง	ส่วนประกอบที่เกี่ยวข้อง
1	com.abc.customerSevicesImpl	Bill_Customers
2	com.abc.financialImpl	Accounts, Benefit_Expense, Pay_Salary, Pay_Supplier
3	com.abc.HumanResourceImpl	Personnel
4	com.abc.InventoryImpl	Inventory, Supply_Office

ตารางที่ ค-4 ผลลัพธ์ความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี

ระบบบริหารงานบริษัทเอบีซี			
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1			
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2			
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	ประเภทความเปลี่ยนแปลง
1	Tax_Revenue	Component	Add-In
2	INF21	Interface	Add-In
3	INF-C10-3	Interface	Add-In
4	Link17	Link	Add-In
5	Benefit_Expense	Component	Modify
6	Bill_Customers	Component	Modify
7	Pay_Supplier	Component	Modify
8	Connector10	Connector	Modify
9	Connector4	Connector	Modify
10	INF-C3-1	Interface	Modify
11	Link20	Link	Modify

ตารางที่ ค-5 ผลลัพธ์รายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี

ระบบบริหารงานบริษัทเอบีซี					
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1					
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2					
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	การเปลี่ยนแปลงคุณสมบัติ	ค่าเดิม	ค่าใหม่
1	Benefit_Expense	Component	Description	Reimburse_Expense	Benefit_Expense

ตารางที่ ค-5 ผลลัพธ์รายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 1 ระบบบริหารงานบริษัทเอบีซี (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	การเปลี่ยนแปลงคุณสมบัติ	ค่าเดิม	ค่าใหม่
2	Bill_Customers	Component	Component Types	Financial Component	Customer Services Component
3	Pay_Supplier	Component	Design Description	Reliability	Functionality, Reliability, Efficiency
4	Connector10	Connector	No of Interfaces	2	3
5	Connector4	Connector	Connector Types	TCPIP Connector	IPTunnels Connector
6	INF-C3-1	Interface	Interface Types	TCPIP Interface	Secure Interface
7	Link20	Interface	Link Pointer	INF14	INF21

ตารางที่ ค-6 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อความต้องการซอฟต์แวร์

ผลกระทบต่อความต้องการซอฟต์แวร์		
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2		
ลำดับที่	ชื่อความต้องการ	ส่วนประกอบที่เกี่ยวข้อง
1	10 - Use for personnel business workflow grouping	PersonnelInterface
2	11 - Use for inventory business workflow grouping	InventoryInterface

ตารางที่ ค-6 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อความต้องการซอฟต์แวร์ (ต่อ)

ลำดับที่	ชื่อความต้องการ	ส่วนประกอบที่เกี่ยวข้อง
3	1 - Use for company account processing balance	Accounts
4	2 - Use for payback expense and automatic accounting balance	Reimburse_Expense
5	3 - Use for salary payment and automatic accounting balance	Pay_Salary
6	4 - Use for supplier payment by export	Pay_Supplier
7	5 - Use for print customer payment slip	Bill_Customers
8	6 - Use for inventory automatic balance	Inventory
9	7 - Use for office stuff and automatic update current of office stock	Supply_Office
10	8 - Use for enquiry employee historical data	Personnel
11	9 - Use for payment business workflow grouping	PaymentInterface

ตารางที่ ค-7 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ

ผลกระทบต่อคุณลักษณะเชิงคุณภาพ		
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2		
ลำดับที่	ชื่อคุณลักษณะเชิงคุณภาพ	ส่วนประกอบที่เกี่ยวข้อง
1	Efficiency	Accounts, Inventory
2	Functionality	Pay_Salary

ตารางที่ ค-7 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ (ต่อ)

ลำดับที่	ชื่อคุณลักษณะเชิงคุณภาพ	ส่วนประกอบที่เกี่ยวข้อง
3	Maintainability	Accounts, InventoryInterface, PaymentInterface, PersonnelInterface, Supply_Office
4	Portability	Accounts, Personnel
5	Reliability	Accounts, Pay_Supplier
6	Usability	Bill_Customers

ตารางที่ ค-8 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี ส่วนผลกระทบต่อชุดคำสั่ง

ผลกระทบต่อชุดคำสั่ง		
ลำดับที่	ชื่อชุดคำสั่ง	ส่วนประกอบที่เกี่ยวข้อง
1	com.abc.financialImpl	Accounts, Bill_Customers, Pay_Salary, PaymentInterface, Reimburse_Expense, Pay_Supplier
2	com.abc.HumanResourceImpl	PersonnelInterface, Personnel
3	com.abc.InventoryImpl	Inventory, InventoryInterface, Supply_Office

ตารางที่ ค-9 ผลลัพธ์ความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์

กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี

ระบบบริหารงานบริษัทเอบีซี			
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1			
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2			
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	ประเภทความเปลี่ยนแปลง
1	InventoryInterface	Component	Add-In
2	PaymentInterface	Component	Add-In
3	PersonnelInterface	Component	Add-In
4	INF21	Interface	Add-In
5	INF22	Interface	Add-In
6	INF23	Interface	Add-In
7	INF24	Interface	Add-In
8	INF25	Interface	Add-In
9	INF26	Interface	Add-In
10	Accounts	Component	Modify
11	Inventory	Component	Modify
12	Personnel	Component	Modify
13	Link10	Link	Modify
14	Link12	Link	Modify
15	Link14	Link	Modify
16	Link15	Link	Modify
17	Link16	Link	Modify
18	Link18	Link	Modify
19	Link19	Link	Modify
20	Link2	Link	Modify
21	Link20	Link	Modify
22	Link4	Link	Modify

ตารางที่ ค-9 ผลลัพธ์ความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	ประเภทความเปลี่ยนแปลง
23	Link5	Link	Modify
24	Link7	Link	Modify
25	Link9	Link	Modify
26	Connector10	Connector	Remove
27	Connector5	Connector	Remove
28	Connector8	Connector	Remove
29	Connector9	Connector	Remove
30	INF02	Interface	Remove
31	INF03	Interface	Remove
32	INF04	Interface	Remove
33	INF13	Interface	Remove
34	INF14	Interface	Remove
35	INF18	Interface	Remove
36	INF20	Interface	Remove
37	INF-C10-1	Interface	Remove
38	INF-C10-2	Interface	Remove
39	INF-C5-1	Interface	Remove
40	INF-C5-2	Interface	Remove
41	INF-C8-1	Interface	Remove
42	INF-C8-2	Interface	Remove
43	INF-C9-1	Interface	Remove
44	INF-C9-2	Interface	Remove
45	Link17	Link	Remove

ตารางที่ ค-10 ผลลัพธ์รายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 2 ระบบบริหารงานบริษัทเอบีซี

ระบบบริหารงานบริษัทเอบีซี					
จาก สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.1					
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : ABC_M1.2					
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	การเปลี่ยนแปลงคุณสมบัติ	ค่าเดิม	ค่าใหม่
1	Accounts	Component	No of Interface	4	1
2	Inventory	Component	No of Interface	3	1
3	Personnel	Component	No of Interface	3	1
4	Link10	Link	Link Pointer	INF-C5-2	INF-C3-2
5	Link12	Link	Link Pointer	INF13	INF25
6	Link14	Link	Link Pointer	INF-C7-2	INF-C6-1
7	Link15	Link	Link Pointer	INF-C8-1	INF-C2-1
8	Link16	Link	Link Pointer	INF-C20	INF-C2-1
9	Link18	Link	Link Pointer	INF-C9-2	INF-C3-2
10	Link19	Link	Link Pointer	INF-C10-1	INF-C6-1
11	Link2	Link	Link Pointer	INF05	INF21
12	Link20	Link	Link Pointer	INF14	INF-C7-2
13	Link4	Link	Link Pointer	INF18	INF23
14	Link5	Link	Link Pointer	INF02	INF22
15	Link7	Link	Link Pointer	INF10	INF24
16	Link9	Link	Link Pointer	INF-C5-1	INF05

ตารางที่ ค-11 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ส่วนผลกระทบต่อความต้องการซอฟต์แวร์

ผลกระทบต่อความต้องการซอฟต์แวร์		
จาก สถาปัตยกรรมซอฟต์แวร์ : AG_v1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : AG_v2		
ลำดับที่	ชื่อความต้องการ	ส่วนประกอบที่เกี่ยวข้อง
1	1 - Game engine development	Game
2	2 - Connector Input Testing	KeyBoardDriver, MouseDriver
3	3 - Connector Display Testing	DisplayDriver
4	4 - Key input testing	KeyBoardDriver, MouseDriver
5	5 - Display output testing	DisplayDriver
6	6 - Online Game Implementation	ConntrollerEngine, DatabaseClient, DatabaseServer

ตารางที่ ค-12 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ

ผลกระทบต่อคุณลักษณะเชิงคุณภาพ		
จาก สถาปัตยกรรมซอฟต์แวร์ : AG_v1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : AG_v2		
ลำดับที่	ชื่อคุณลักษณะเชิงคุณภาพ	ส่วนประกอบที่เกี่ยวข้อง
1	Efficiency	ConntrollerEngine, DatabaseClient, DisplayDriver, KeyBoardDriver, MouseDriver, DatabaseServer

ตารางที่ ค-12 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ (ต่อ)

ลำดับที่	ชื่อคุณลักษณะเชิงคุณภาพ	ส่วนประกอบที่เกี่ยวข้อง
2	Functionality	ConntrollerEngine, DatabaseClient, DisplayDriver, Game, KeyBoardDriver, MouseDriver, DatabaseServer
3	Maintainability	ConntrollerEngine, DatabaseClient, DatabaseServer
4	Portability	ConntrollerEngine, DatabaseClient, DatabaseServer
5	Reliability	DisplayDriver, KeyBoardDriver, MouseDriver
6	Usability	DisplayDriver, KeyBoardDriver, MouseDriver

ตารางที่ ค-13 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ ส่วนผลกระทบต่อชุดคำสั่ง

ผลกระทบต่อชุดคำสั่ง		
จาก สถาปัตยกรรมซอฟต์แวร์ : AG_v1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : AG_v2		
ลำดับที่	ชื่อชุดคำสั่ง	ส่วนประกอบที่เกี่ยวข้อง
1	com.arcade.core.Engine	ConntrollerEngine, Game
2	com.arcade.Input	KeyBoardDriver, MouseDriver
3	com.arcade.Output	DisplayDriver

ตารางที่ ค-14 ผลลัพธ์ความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์

ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์			
จาก สถาปัตยกรรมซอฟต์แวร์ : AG_v1			
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : AG_v2			
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	ประเภทความเปลี่ยนแปลง
1	ConntrollerEngine	Component	Add-In
2	DatabaseClient	Component	Add-In
3	DatabaseServer	Component	Add-In
4	ConnectorCE	Connector	Add-In
5	ConnectorDBClient	Connector	Add-In
6	ConnectorDBServer	Connector	Add-In
7	CCEInf1	Interface	Add-In
8	CCEInf2	Interface	Add-In
9	CDBCInf1	Interface	Add-In
10	CDBCInf2	Interface	Add-In
11	CDBSInf1	Interface	Add-In
12	CDBSInf2	Interface	Add-In
13	CEInf1	Interface	Add-In
14	CEInf2	Interface	Add-In
15	CEInf3	Interface	Add-In
16	DBCInf1	Interface	Add-In
17	DBCInf2	Interface	Add-In
18	DBSInf1	Interface	Add-In
19	Link1-1	Link	Add-In
20	Link1-2	Link	Add-In
21	Link1-3	Link	Add-In
22	Link1-4	Link	Add-In

ตารางที่ ค-14 ผลลัพธ์ความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์ (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ประเภท ส่วนประกอบ	ประเภทความ เปลี่ยนแปลง
23	Link1-5	Link	Add-In
24	Link6	Link	Add-In
25	Link7	Link	Add-In
26	Link8	Link	Add-In
27	USBInf1	Interface	Modify
28	Link1	Link	Remove
29	Link2	Link	Remove

ตารางที่ ค-15 ผลลัพธ์รายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 3 ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์

ระบบเกมส์เครื่องเล่นและเกมส์ออนไลน์					
จาก สถาปัตยกรรมซอฟต์แวร์ : AG_v1					
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : AG_v2					
ลำดับที่	ชื่อส่วนประกอบ	ประเภท ส่วนประกอบ	การ เปลี่ยนแปลง คุณสมบัติ	ค่าเดิม	ค่าใหม่
1	USBInf1	Interface	Description	USBInfInf1	USBInf1

ตารางที่ ค-16 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ส่วนผลกระทบต่อความต้องการซอฟต์แวร์

ผลกระทบต่อความต้องการซอฟต์แวร์		
จาก สถาปัตยกรรมซอฟต์แวร์ : US_v1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : US_v2		
ลำดับที่	ชื่อความต้องการ	ส่วนประกอบที่เกี่ยวข้อง
1	1 - Development Seminar UI Component	SeminarManagementUI

ตารางที่ ค-16 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ส่วนผลกระทบต่อความต้องการซอฟต์แวร์ (ต่อ)

ลำดับที่	ชื่อความต้องการ	ส่วนประกอบที่เกี่ยวข้อง
2	2 - Development Student UI Component	StudentAdministrationUI
3	3 - Development Facilities Entry Module	Facilities
4	4 - Development Student Entry Module	Student
5	5 - Development Seminar Entry Module	Seminar
6	6 - Development Schedule Entry Module	Schedule
7	7 - Development Common Persistence Entry Module	Persistence, UniversityDatabase1
8	8 - Development Personal Security Module	UniversityDatabase2

ตารางที่ ค-17 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ

ผลกระทบต่อคุณลักษณะเชิงคุณภาพ		
จาก สถาปัตยกรรมซอฟต์แวร์ : US_v1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : US_v2		
ลำดับที่	ชื่อคุณลักษณะเชิงคุณภาพ	ส่วนประกอบที่เกี่ยวข้อง
1	Efficiency	Facilities, Persistence, Seminar, Student, UniversityDatabase1, UniversityDatabase2, Schedule

ตารางที่ ค-17 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์ กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ส่วนผลกระทบต่อคุณลักษณะเชิงคุณภาพ (ต่อ)

ลำดับที่	ชื่อคุณลักษณะเชิงคุณภาพ	ส่วนประกอบที่เกี่ยวข้อง
2	Functionality	Facilities, LoggerSpooling, Persistence, Seminar, SeminarManagementUI, Student, StudentAdministrationUI, UniversityDatabase1, Schedule
3	Maintainability	Facilities, LoggerSpooling, Persistence, Seminar, Student, UniversityDatabase1, UniversityDatabase2, Schedule
4	Reliability	Facilities, LoggerSpooling, Persistence, Seminar, Student, UniversityDatabase1, UniversityDatabase2, Schedule
5	Usability	SeminarManagementUI, StudentAdministrationUI

ตารางที่ ค-18 ผลลัพธ์การตรวจจับผลกระทบของการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย ส่วนผลกระทบต่อชุดคำสั่ง

ผลกระทบต่อชุดคำสั่ง		
จาก สถาปัตยกรรมซอฟต์แวร์ : US_v1		
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : US_v2		
ลำดับที่	ชื่อชุดคำสั่ง	ส่วนประกอบที่เกี่ยวข้อง
1	com.university.entry.businessProcess	Facilities, Seminar, Student, Schedule
2	com.university.entry.common.Utils	Persistence, UniversityDatabase1, UniversityDatabase2, LoggerSpooling
3	com.university.entry.UIService	SeminarManagementUI, StudentAdministrationUI

ตารางที่ ค-19 ผลลัพธ์ความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย

ระบบการลงทะเบียนมหาวิทยาลัย			
จาก สถาปัตยกรรมซอฟต์แวร์ : US_v1			
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : US_v2			
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	ประเภทความเปลี่ยนแปลง
1	LoggerSpooling	Component	Add-In
2	UniversityDatabase2	Component	Add-In
3	DatabaseLoadBalancer	Connector	Add-In
4	LogConnector	Connector	Add-In
5	iDBL1	Interface	Add-In
6	iDBL2	Interface	Add-In
7	iDBL3	Interface	Add-In

ตารางที่ ค-19 ผลลัพธ์ความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	ประเภทความเปลี่ยนแปลง
8	iLogCon1	Interface	Add-In
9	iLogCon2	Interface	Add-In
10	iLogSpo1	Interface	Add-In
11	iPer3	Interface	Add-In
12	iPerCon3	Interface	Add-In
13	iPerCon4	Interface	Add-In
14	iPerCon5	Interface	Add-In
15	iSem3	Interface	Add-In
16	iSemC1-3	Interface	Add-In
17	iSemC1-4	Interface	Add-In
18	iSemC1-5	Interface	Add-In
19	iUDB2	Interface	Add-In
20	I15IDBL2IUDB	Link	Add-In
21	I16IDBL3IUDB2	Link	Add-In
22	I17iPer3iLogCon1	Link	Add-In
23	I18iLogCon2iLongSpo1	Link	Add-In
24	Persistence	Component	Modify
25	Seminar	Component	Modify
26	UniversityDatabase1	Component	Modify
27	PersistenceConnector	Connector	Modify
28	SeminarConnetor	Connector	Modify
29	I12IDataCon2IDBL1	Link	Modify
30	I14IStuCon2ISem3	Link	Modify
31	I3ISemCon1-3IStu1	Link	Modify
32	I4ISemCon1-4ISem1	Link	Modify

ตารางที่ ค-19 ผลลัพธ์ความเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	ประเภทความเปลี่ยนแปลง
33	I5ISemCon1-5ISch1	Link	Modify
34	I7IStu2IPerCon3	Link	Modify
35	I8ISem2IPerCon4	Link	Modify
36	I9ISch2IPerCon5	Link	Modify

ตารางที่ ค-20 ผลลัพธ์รายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย

ระบบการลงทะเบียนมหาวิทยาลัย					
จาก สถาปัตยกรรมซอฟต์แวร์ : US_v1					
ไปยัง สถาปัตยกรรมซอฟต์แวร์ : US_v2					
ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	การเปลี่ยนแปลงคุณสมบัติ	ค่าเดิม	ค่าใหม่
1	Persistence	Component	No of Interface	2	3
2	Seminar	Component	No of Interface	2	3
3	UniversityDatabase1	Component	Description	University Database	University Database1
4	PersistenceConnector	Connector	No of Interface	2	5
5	SeminarConnetor	Connector	No of Interface	2	5

ตารางที่ ค-20 ผลลัพธ์รายละเอียดการเปลี่ยนแปลงสถาปัตยกรรมซอฟต์แวร์
กรณีทดสอบที่ 4 ระบบการลงทะเบียนมหาวิทยาลัย (ต่อ)

ลำดับที่	ชื่อส่วนประกอบ	ประเภทส่วนประกอบ	การเปลี่ยนแปลงคุณสมบัติ	ค่าเดิม	ค่าใหม่
6	I12IDataCon2IDBL1	Link	Description	I12IData Con2IUDB	I12IData Con2IDBL1
		Link	Link Pointer	iUDB	iDBL1
7	I14IStuCon2ISem3	Link	Description	I14IStu Con2ISem1	I14IStu Con2ISem3
		Link	Link Pointer	iSem1	iSem3
8	I3ISemCon1-3IStu1	Link	Link Pointer	iSemC1-2	iSemC1-3
9	I4ISemCon1-4ISem1	Link	Link Pointer	iSemC1-2	iSemC1-4
10	I5ISemCon1-5ISch1	Link	Link Pointer	iSemC1-2	iSemC1-5
11	I7IStu2IPerCon3	Link	Description	I7IStu2 IPerCon1	I7IStu2 IPerCon3
		Link	Link Pointer	iPerCon1	iPerCon3
12	I8ISem2IPerCon4	Link	Description	I8ISem2 IPerCon1	I8ISem2 IPerCon4
		Link	Link Pointer	iPerCon1	iPerCon4
13	I9ISch2IPerCon5	Link	Description	I9ISch2 IPerCon1	I9ISch2 IPerCon5
		Link	Link Pointer	iPerCon1	iPerCon5

ประวัติผู้เขียนวิทยานิพนธ์

นายอาทิตย์ อุดมสมฤดี เกิดวันศุกร์ที่ 6 กรกฎาคม พ.ศ. 2527 ที่จังหวัดนครสวรรค์ สำเร็จการศึกษาในหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยกรุงเทพฯ เมื่อปีการศึกษา 2549 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2554