การวิเคราะห์และออกแบบนวัตกรรมเทคโนโลยีธุรกิจสมัยใหม่

Innovation Technology Modern Business Design and Analysis

รหัสวิชา 1207 204

ผู้ช่วยศาสตราจารย์ศศิธร แก้วมั่น

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ

มหาวิทยาลัยมหาสารคาม

มหาวิทยาลัยมหาสารคาม

Mahasarakham University

การวิเคราะห์และออกแบบนวัตกรรมเทคโนโลยีธุรกิจสมัยใหม่
Innovation Technology Modern Business Design and Analysis

จัดทำโดย : ผู้ช่วยศาสตราจารย์ศศิธร แก้วมั่น
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม
ตำบลขามเรียง อำเภอกันทรวิชัย จังหวัดมหาสารคาม 44150 ประเทศไทย
sasitorn.k@msu.ac.th

ข้อมูลทางบรรณานุกรมของหอสมุดแห่งชาติ

ศศิธร แก้วมั่น.
การวิเคราะห์และออกแบบนวัตกรรมเทคโนโลยีธุรกิจสมัยใหม่ = Innovation technology modern business design and analysis.-- มหาสารคาม : โรงพิมพ์พีคัลเลอร์, 2566.
189 หน้า.
1. การวิเคราะห์ระบบ. 2. การออกแบบระบบ. 3. นวัตกรรมทางเทคโนโลยี. I. ชื่อเรื่อง.
004.21
ISBN: 978-616-608-080-3

สงวนลิขสิทธิ์โดย © ผู้ช่วยศาสตราจารย์ศศิธร แก้วมั่น

พิมพ์ครั้งที่ : 1
ปีที่พิมพ์ : 2566
จำนวน : 200 เล่ม
ราคา : 239 บาท

พิสูจน์อักษร : ผู้ช่วยศาสตราจารย์ภัทธิรา สุวรรณโค
ออกแบบปก : นางสาวจันทร์สุดา การดี
พิมพ์ที่ : โรงพิมพ์พีคัลเลอร์
23/14 ถนนศรีสวัสดิ์ดำเนิน ตำบลตลาด อำเภอเมือง จังหวัดมหาสารคาม 44000

คำนำ

ตำราฉบับนี้จัดทำขึ้นเพื่อใช้ประกอบการเรียนการสอน รหัสวิชา 1207204 การวิเคราะห์และออกแบบนวัตกรรมเทคโนโลยีธุรกิจสมัยใหม่ (Design and Analysis of Innovation Technology Modern Business) มุ่งเน้นให้นิสิตสามารถเข้าใจแนวความคิดพื้นฐาน และการนำหลักการวิเคราะห์และออกแบบระบบไปใช้ประโยชน์และแก้ไขปัญหาเกี่ยวกับการพัฒนาระบบสารสนเทศ ตามคำอธิบายรายวิชาโดยมีเนื้อหาประกอบไปด้วย แนวคิด ระบบและรูปแบบของนวัตกรรมและเทคโนโลยีเพื่อธุรกิจสมัยใหม่ วัฏจักรของการพัฒนาระบบ การสืบค้นและวิเคราะห์ความต้องการของผู้ใช้นวัตกรรมเทคโนโลยี แผนภาพการไหลของข้อมูล พจนานุกรมข้อมูล การอธิบายการประมวลผล การออกแบบแฟ้มข้อมูล ฐานข้อมูล การออกแบบส่วนติดต่อกับผู้ใช้ การพัฒนา การทดสอบและการนำนวัตกรรมเทคโนโลยีไปใช้งานธุรกิจสมัยใหม่ การจัดทำเอกสาร การฝึกปฏิบัติกรณีศึกษาการวิเคราะห์และออกแบบนวัตกรรมเพื่อธุรกิจสมัยใหม่

ตำรานี้สำเร็จลงได้ด้วยการได้รับความร่วมมือจากนิสิต และผู้เกี่ยวข้อง หวังเป็นอย่างยิ่งว่าจะเกิดประโยชน์ต่อไปสำหรับนิสิตที่เรียนในรายวิชานี้และผู้สนใจ

ผู้ช่วยศาสตราจารย์ศศิธร แก้วมั่น

สารบัญ

สารบัญ .... ก

สารบัญภาพ .... ง

สารบัญตาราง .... ช

บทที่ 1 ความรู้เบื้องต้น 1

ความหมายของระบบสารสนเทศ 1

ความหมายของนวัตกรรม 9

ความหมายของเทคโนโลยี 11

ความหมายของธุรกิจสมัยใหม่ 13

สรุปท้ายบทที่ 1 19

คำถามท้ายบทที่ 1 21

บทที่ 2 วงจรของการพัฒนาระบบ 22

วงจรการพัฒนาระบบ (System Development Life Cycle: SDLC) 22

แบบจำลองของวงจรการพัฒนาระบบ (System Development Life Cycle Model : SDLC Model) 28

ระเบียบวิธีการพัฒนาระบบแบบดั้งเดิม (Traditional System Development Methodology).......... 33

ระเบียบวิธีการพัฒนาซอฟต์แวร์แบบเอจายล์ (Agile Software Development Methodology)......... 42

เครื่องมือสนับสนุนกระบวนการพัฒนาระบบ 56

สรุปท้ายบทที่ 2 60

คำถามท้ายบทที่ 2 62

บทที่ 3 การวิเคราะห์ความต้องการของผู้ใช้ 63

การรวบรวมความต้องการของผู้ใช้งาน 64

การวิเคราะห์ความต้องการของระบบ 78

ข้อกำหนดเกี่ยวกับความต้องการของระบบ 80

สารบัญ (ต่อ)

สรุปท้ายบทที่ 3 82

คำถามท้ายบทที่ 3 84

บทที่ 4 แผนภาพกระแสข้อมูล 85

ความรู้เกี่ยวกับแผนภาพกระแสข้อมูล 85

สัญลักษณ์แผนภาพกระแสข้อมูล 86

กฎที่ใช้ในการเขียนแผนภาพกระแสข้อมูล 92

ประเภทของแผนภาพกระแสข้อมูล 95

ประโยชน์ของการสร้างแผนภาพกระแสข้อมูล 96

สรุปท้ายบทที่ 4 98

คำถามท้ายบทที่ 4 99

บทที่ 5 การออกแบบฐานข้อมูล 100

การจัดการข้อมูลแบบดั้งเดิม 101

องค์ประกอบของระบบฐานข้อมูล 105

แบบจำลองข้อมูล (Data Model) 106

ระบบจัดการฐานข้อมูล (Database Management System : DBMS) 112

ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) 114

แบบจำลอง E-R (Entity-Relational Modeling) 117

กระบวนการในการพัฒนาแผนภาพ E-R ในองค์กรธุรกิจสมัยใหม่ 124

การเขียนคำสั่ง SQL จัดการฐานข้อมูล 125

สรุปท้ายบทที่ 5 133

คำถามท้ายบทที่ 5 134

บทที่ 6 การออกแบบส่วนติดต่อผู้ใช้ 135

องค์ประกอบของส่วนติดต่อผู้ใช้ 136

สารบัญ (ต่อ)

ระบบจัดการสื่อประสานกับผู้ใช้ (User Interface Management System : UIMS)........................ 138
ประเภทของส่วนติดต่อผู้ใช้ 140
หลักการออกแบบส่วนติดต่อผู้ใช้ 141
การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface)... 143
การออกแบบส่วนติดต่อผู้ใช้สมัยใหม่.. 147
สรุปท้ายบทที่ 6 153
คำถามท้ายบทที่ 6 154
บทที่ 7 ระบบขายหน้าร้าน (Point of Sale System).. 155
การรวบรวมความต้องการใช้งานของผู้ใช้งาน 155
การวิเคราะห์ความต้องการของระบบ.. 156
การสร้างแผนภาพกระแสข้อมูล 157
คำอธิบายกระบวนการแผนภาพกระแสข้อมูล 170
การสร้างแผนภาพ E-R.. 174
โครงสร้างฐานข้อมูล... 177
สรุปท้ายบทที่ 7 179
คำถามท้ายบทที่ 7 180
บรรณานุกรม 181
ดัชนี.. 184

สารบัญภาพ

ภาพที่ 1.1 อธิบายประเภทนวัตกรรมด้วย Innovation Matrix 9
ภาพที่ 1.2 ความแตกต่างระหว่างธุรกิจรูปแบบ PIPE กับรูปแบบ PLATFORM 15
ภาพที่ 2.1 วงจรการพัฒนาระบบ SDLC 23
ภาพที่ 2.2 แบบจำลองน้ำตก 29
ภาพที่ 2.3 แบบจำลองปรับปรุงน้ำตกที่วนซ้ำได้ 30
ภาพที่ 2.4 แบบจำลองแบบเพิ่มผลลัพธ์ 31
ภาพที่ 2.5 แบบจำลองแบบเกลียว 32
ภาพที่ 2.6 แบบจำลองการสร้างโปรแกรมต้นแบบในระยะเริ่มต้น 33
ภาพที่ 2.7 ระเบียบวิธีการพัฒนาแอปพลิเคชันแบบรวดเร็ว 36
ภาพที่ 2.8 วงจรการพัฒนาซอฟต์แวร์แบบเอจายล์ 43
ภาพที่ 2.9 ขั้นตอนการทำงานของ SCRUM 50
ภาพที่ 3.1 แหล่งข้อมูลของความต้องการของระบบ 65
ภาพที่ 3.2 วิธีการรวบรวมความต้องการของผู้ใช้ 67
ภาพที่ 4.1 สัญลักษณ์แผนภาพกระแสข้อมูล 86
ภาพที่ 4.2 ตัวอย่างการใช้สัญลักษณ์การประมวลผล (Process) 87
ภาพที่ 4.3 ตัวอย่างสัญลักษณ์กระแสข้อมูลอินพุต 88
ภาพที่ 4.4 ตัวอย่างสัญลักษณ์กระแสข้อมูลเอาต์พุต 88
ภาพที่ 4.5 ตัวอย่างสัญลักษณ์เอ็กซ์เทอร์นัลเอนทิตี 89
ภาพที่ 4.6 เอ็กซ์เทอร์นัลเอนทิตีของลูกค้า และสัญลักษณ์การทำซ้ำของลูกค้า 90
ภาพที่ 4.7 ตัวอย่างสัญลักษณ์ดาตาสโตร์ (Data Stores) 90
ภาพที่ 4.8 แสดงโปรเซสที่มีการอินพุตจากดาตาสโตร์ 91
ภาพที่ 4.9 แสดงโปรเซสที่มีการเอาต์พุตเข้าไปยังดาตาสโตร์ 91
ภาพที่ 4.10 แสดงโปรเซสที่มีการจัดการข้อมูลในดาตาสโตร์ทั้งแบบอินพุตและเอาต์พุต 91
ภาพที่ 4.11 แสดงสัญลักษณ์เรียลไทม์ลิงก์ 92
ภาพที่ 5.1 ระบบแฟ้มข้อมูล 102
ภาพที่ 5.2 ระบบจัดการฐานข้อมูล 104

สารบัญภาพ (ต่อ)

ภาพที่ 5.3 แบบจำลองฐานข้อมูลลำดับชั้น 108
ภาพที่ 5.4 แบบจำลองฐานข้อมูลเครือข่าย 109
ภาพที่ 5.5 แบบจำลองฐานข้อมูลเชิงสัมพันธ์ 110
ภาพที่ 5.6 แบบจำลองฐานข้อมูลเชิงวัตถุ 111
ภาพที่ 5.7 แบบจำลองฐานข้อมูลแบบมัลติไดเมนชั่น 112
ภาพที่ 5.8 สัญลักษณ์เอนทิตี 119
ภาพที่ 5.9 Strong Entity และ Weak Entity 119
ภาพที่ 5.10 เอนทิตี Employee ซึ่งประกอบด้วยแอตทริบิวต์ต่างๆ 120
ภาพที่ 5.11 เปรียบเทียบระหว่าง Simple Attribute และ Composite Attribute 120
ภาพที่ 5.12 Single-Valued Attribute และ Multivalued Attribute 121
ภาพที่ 5.13 แอตทริบิวต์ age เป็น Derived Attribute 121
ภาพที่ 5.14 แสดงการอธิบายความสัมพันธ์ระหว่างอาจารย์และรายวิชาที่สอน 122
ภาพที่ 5.15 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (1:1) 122
ภาพที่ 5.16 ความสัมพันธ์ระหว่างเอนทิตีแบบ One-to-Many 123
ภาพที่ 5.17 ความสัมพันธ์ระหว่างเอนทิตีแบบ Many-to-Many 123
ภาพที่ 5.18 ความสัมพันธ์ระหว่างเอนทิตีแบบ One-to-Many ที่แปลงจาก Many-to-Many 124
ภาพที่ 6.1 องค์ประกอบของสื่อประสานกับผู้ใช้ 136
ภาพที่ 6.2 หน้าจอการโต้ตอบด้วยคำสั่ง 144
ภาพที่ 6.3 เมนูแสดง Pull-down Menu 145
ภาพที่ 6.4 เมนูแสดง Pop-up Menu 145
ภาพที่ 6.5 การโต้ตอบด้วยแบบฟอร์ม 146
ภาพที่ 6.6 การโต้ตอบด้วยการทำงานเชิงวัตถุ 147
ภาพที่ 6.7 การโต้ตอบด้วยภาษามนุษย์ 147
ภาพที่ 6.8 โครงสร้างการทำงานของโปรแกรมกราฟิก 148
ภาพที่ 6.9 มัลติมีเดียและไฮเปอร์เท็กซ์ 150
ภาพที่ 6.10 ระบบความเป็นจริงเสมือน 151

สารบัญภาพ (ต่อ)

ภาพที่ 7.1 แผนภาพบริบทของระบบ POS....160
ภาพที่ 7.2 แผนภาพระดับที่ 0 ของระบบ POS....161
ภาพที่ 7.3 แผนภาพระดับที่ 1 ของกระบวนการที่ 1 จัดการสมาชิก....162
ภาพที่ 7.4 แผนภาพระดับที่ 1 ของกระบวนการที่ 2 เข้าสู่ระบบ....163
ภาพที่ 7.5 แผนภาพระดับที่ 1 ของกระบวนการที่ 3 ค้นหาข้อมูลสินค้า....164
ภาพที่ 7.6 แผนภาพระดับที่ 1 ของกระบวนการที่ 4 สั่งซื้อสินค้า....165
ภาพที่ 7.7 แผนภาพระดับที่ 1 ของกระบวนการที่ 5 จัดการสินค้า และระดับที่ 2 ของกระบวนการที่ 5.1 เพิ่มสินค้า....166
ภาพที่ 7.8 แผนภาพระดับที่ 1 ของกระบวนการที่ 6 ตรวจสอบการสั่งซื้อสินค้าของลูกค้า....167
ภาพที่ 7.9 แผนภาพระดับที่ 1 ของกระบวนการที่ 7 ตรวจสอบสินค้าคงเหลือ....168
ภาพที่ 7.10 แผนภาพระดับที่ 1 ของกระบวนการที่ 8 ตรวจสอบสินค้าคงเหลือ....169
ภาพที่ 7.11 แผนภาพ E-R ระบบขายหน้าร้าน POS....175

สารบัญตาราง

ตารางที่ 2.1 หลักการพื้นฐานของการพัฒนาระบบแบบเอจายล์ 5 ประการ 43
ตารางที่ 2.2 การเปรียบเทียบระหว่างการจัดการโครงการแบบดั้งเดิมและแบบเอจายล์ 46
ตารางที่ 3.1 การเปรียบเทียบข้อดีและข้อจำกัดของคำถามแต่ละรูปแบบ 75
ตารางที่ 3.2 ตัวอย่างข้อความที่เหมาะสมสำหรับการจัดทำ SRS 81
ตารางที่ 3.3 ตัวอย่าง SRS ของระบบซื้อขายสินค้าออนไลน์ 82
ตารางที่ 4.1 กฎในการสร้างแผนภาพกระแสข้อมูล 93
ตารางที่ 4.2 ข้อแตกต่างระหว่างแผนภาพกระแสข้อมูลแบบตรรกะกับแบบกายภาพ 96
ตารางที่ 7.1 คำสำคัญที่น่าจะเป็นองค์ประกอบของแผนภาพกระแสข้อมูล 158
ตารางที่ 7.2 ความสัมพันธ์ขององค์ประกอบต่างๆ สำหรับแต่ละกระบวนการทำงาน 159
ตารางที่ 7.3 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 1 จัดการสมาชิก 170
ตารางที่ 7.4 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 2 เข้าสู่ระบบ 170
ตารางที่ 7.5 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 3 ค้นหาข้อมูลสินค้า 171
ตารางที่ 7.6 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 4 สั่งซื้อสินค้า 171
ตารางที่ 7.7 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 5 จัดการสินค้า 172
ตารางที่ 7.8 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 6 ตรวจสอบการสั่งซื้อ 172
ตารางที่ 7.9 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 7 ตรวจสอบสินค้าคงเหลือ 173
ตารางที่ 7.10 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 7 แสดงรายงานยอดสั่งซื้อแต่ละวัน เดือน ปี 173
ตารางที่ 7.11 กำหนดสิทธิ์เข้าใช้งานสมาชิก (UserRole) 177
ตารางที่ 7.12 สมาชิก (UserDetail) 177
ตารางที่ 7.13 ประเภทสินค้า (ProductType) 177
ตารางที่ 7.14 สินค้า (Products) 178
ตารางที่ 7.15 การสั่งซื้อสินค้า (Orders) 178
ตารางที่ 7.16 รายละเอียดการสั่งซื้อสินค้า (OrderDetail) 178

บทที่ 1
ความรู้เบื้องต้น

ปัจจุบันระบบสารสนเทศได้เข้ามามีบทบาทต่อมนุษย์เป็นอย่างมาก ทั้งในการดำเนินชีวิตประจำวันและการประกอบอาชีพ ดังนั้น การพัฒนาและการใช้งานระบบสารสนเทศให้เหมาะสมกับตนเองและองค์กรจึงเป็นสิ่งจำเป็นเพื่อเพิ่มศักยภาพในการแข่งขันสำหรับโลกในยุคโลกาภิวัฒน์

เนื่องจากปัจจุบันธุรกิจไม่ได้จำกัดเป้าหมายอยู่ที่การตอบสนองความต้องการของกลุ่มเป้าหมายในภูมิภาคหรือกลุ่มเป้าหมายเดิมเท่านั้น แต่ได้ขยายครอบคลุมไปทั่วโลก ผู้บริหารของแต่ละองค์กรจึงจำเป็นต้องให้ความสำคัญแก่ระบบสารสนเทศที่มีคุณภาพ เพื่อให้การดำเนินงานขององค์กรเป็นไปอย่างราบรื่น รวมไปถึงการตัดสินใจของผู้บริหารหรือการกำหนดกลยุทธ์ต่าง ๆ ขององค์กรให้อยู่บนพื้นฐานของข้อมูลและสารสนเทศให้ครบถ้วนสมบูรณ์ที่สุด

จากที่ได้กล่าวมาแล้วข้างต้น เนื้อหาในบทนี้จึงได้แบ่งออกเป็น 4 หัวข้อหลักคือ ระบบสารสนเทศ นวัตกรรม เทคโนโลยี และธุรกิจสมัยใหม่

ความหมายของระบบสารสนเทศ

ระบบสารสนเทศ (Information System) ประกอบด้วยคำ 2 คำคือ (โอภาส เอี่ยมสิริวงศ์, 2558; ศรีไพร ศักดิ์รุ่งพงศากุล และเจษฎาพร ยุทธนวิบูลย์ชัย, 2549)

1) ระบบ (System) หมายถึง กลุ่มขององค์ประกอบหลาย ๆ องค์ประกอบที่ทำงานร่วมกันเพื่อตอบสนอง วัตถุประสงค์อย่างเดียวกัน

2) สารสนเทศ (Information) หมายถึง ข้อมูลที่ผ่านการประมวลผลเพื่อนำไปใช้ประโยชน์ในด้านการกระบวนการสร้างจัดการอย่างใดอย่างหนึ่ง โดยกระบวนการสร้างสารสนเทศสามารถแสดงได้

2.1) ข้อมูล (Data) เป็นข้อเท็จจริงเกี่ยวกับเหตุการณ์ที่เกิดขึ้น ซึ่งเป็นข้อมูลดิบ (Raw Data) ที่ยังไม่ผ่านการประมวลผล จึงยังไม่มีความหมายในการนำไปใช้งาน เช่น สินค้าที่ลูกค้าสั่งซื้อ คะแนนสอบแต่ละรายวิชา และเงินเดือนของพนักงานในองค์กร เป็นต้น

2.2) การประมวลผล (Processing) เป็นวิธีการเปลี่ยนแปลงข้อมูลที่มีอยู่แล้วให้เป็นสารสนเทศตามที่ผู้ใช้ต้องการ เช่น วิธีการคำนวณรายได้จากการขายสินค้า วิธีการประมวลผลการเรียนของนักศึกษาในแต่ละรายวิชา และวิธีการคำนวณเงินได้สุทธิของพนักงานในแต่ละเดือน เป็นต้น

2.3) สารสนเทศ (Information) เป็นผลลัพธ์จากการประมวลผลข้อมูลตามวัตถุประสงค์ของการใช้งาน เช่น รายได้จากการขายสินค้า ผลการเรียนของนักศึกษาในแต่ละรายวิชา และเงินได้สุทธิของพนักงานในแต่ละเดือน เป็นต้น

2.4) ผลย้อนกลับ (Feedback) เป็นผลที่เกิดจากการใช้งานสารสนเทศนั้น ๆ เช่น รูปแบบการนำเสนอ รายงานที่ไม่ครอบคลุมกับสถานการณ์ปัจจุบัน ข้อมูลที่ไม่ทันสมัย หรือความผิดพลาดในการประมวลผลข้อมูล เป็นต้น ซึ่งจะเป็นประโยชน์ในการนำมาปรับปรุงข้อมูล สารสนเทศ หรือการ ประมวลผลให้มีประสิทธิภาพมากยิ่งขึ้น

จากที่ได้กล่าวไว้แล้วข้างต้น ระบบสารสนเทศ จึงหมายถึงกลุ่มขององค์ประกอบทางด้านสารสนเทศที่มีความสัมพันธ์กันและสามารถทำงานร่วมกันเพื่อตอบสนองความต้องการใช้งานในแต่ละองค์กรภายในระบบสารสนเทศหนึ่ง ๆ ยังสามารถแบ่งได้เป็นระบบย่อย (Subsystem) อีกหลายระบบที่มี ความสัมพันธ์กัน โดยแต่ละระบบย่อยจะมีวัตถุประสงค์ในการใช้งานที่แตกต่างกันและสามารถทำงานได้สมบูรณ์ในตัวเอง

2. ขอบเขตของระบบสารสนเทศและสิ่งแวดล้อม

ระบบสารสนเทศของแต่ละองค์กรแม้จะมีชื่อเรียกที่เหมือนกัน แต่อาจจะมีฟังก์ชันการทำงานที่แตกต่างกันได้ ทั้งนี้ขึ้นอยู่กับขอบเขตของระบบ (System Boundary) ซึ่งเป็นสิ่งสำคัญในการระบุถึงความสามารถในภาพรวมของทั้งระบบ (โอภาส เอี่ยมสิริวงศ์, 2558; ศรีไพร ศักดิ์รุ่งพงศากุล และเจษฎาพร ยุทธนวิบูลย์ชัย, 2549)

ในกรณีที่เป็นระยะเริ่มต้นของการพัฒนาระบบ กิจกรรมสำคัญสำหรับนักวิเคราะห์ระบบคือการกำหนดถึงขอบเขตของระบบให้ครบถ้วนและชัดเจนที่สุด เพื่อให้เกิดความเข้าใจร่วมกันระหว่างทีมนักพัฒนาระบบและผู้ที่เกี่ยวข้องกับระบบ นั่นหมายความว่า ความสามารถใดที่ไม่ได้ระบุไว้ในขอบเขตของระบบก็ยังไม่จำเป็นต้องพัฒนาในส่วนนั้น

อีกประเด็นเนื่องจากทุกสรรพสิ่งในโลกจะมีปฏิสัมพันธ์กับสิ่งแวดล้อมเสมอไม่ว่าจะเป็นสิ่งแวดล้อมภายในและสิ่งแวดล้อมภายนอกก็ตาม สำหรับระบบสารสนเทศก็เช่นเดียวกัน การจำแนกว่าสิ่งแวดล้อมใดเป็นสิ่งแวดล้อมภายในหรือสิ่งแวดล้อมภายนอกจะอาศัยขอบเขตของระบบเป็นตัวกำหนด สิ่งแวดล้อมในที่นี้หมายถึงปัจจัยที่มีผลต่อการทำงานและการคงอยู่ของระบบสารสนเทศ โดยแบ่งออกเป็น 2 กลุ่มหลัก

2.1 สิ่งแวดล้อมภายในระบบ (Internal Environment) หมายถึง ปัจจัยใด ๆ ที่อยู่ภายในระบที่ส่งผลต่อการทำงานและการคงอยู่ของระบบ เช่น โครงสร้างองค์กร บุคลากรในองค์กร อุปกรณ์ เครื่องมือ วัฒนธรรมขององค์กร ขั้นตอนการทำงาน และความขัดแย้งระหว่างหน่วยงานย่อยในองค์กร เป็นต้น

2.2 สิ่งแวดล้อมภายนอกระบบ (External Environment) หมายถึง ปัจจัยใด ๆ ที่อยู่ภายนอกระบบ ซึ่งส่งผลต่อการทำงานและการคงอยู่ของระบบ เช่น ลูกค้า คู่แข่ง ผู้ผลิตวัตถุดิบ องค์กรที่มีความร่วมมือกัน สภาพเศรษฐกิจ กฎหมายและระเบียบข้อบังคับต่าง ๆ และเทคโนโลยีสารสนเทศที่เปลี่ยนแปลงไป เป็นต้น

ระบบสารสนเทศก็เหมือนกับระบบอื่น ๆ ที่เราได้พบเห็นอยู่ทั่วไป ยกตัวอย่างเช่น ระบบในร่างกายของมนุษย์ หน่วยที่เล็กที่สุดคือ เซลล์ ดังนั้น ถ้ามองว่าเซลล์หนึ่งเซลล์เป็นหนึ่งระบบก็ได้ เพราะเซลล์มีองค์ประกอบ และหน้าที่ที่ชัดเจน หรือมองว่ากลุ่มของเซลล์ที่มารวมกันเป็นอวัยวะใดอวัยวะหนึ่งเป็นหนึ่งระบบก็ได้ และในทำนองเดียวกันมองว่าร่างกายของมนุษย์ที่ประกอบด้วยอวัยวะทั้งหมดที่ทำงานสัมพันธ์กันเป็นหนึ่งระบบก็ได้

ดังนั้น สิ่งสำคัญเป็นอันดับแรกของการพัฒนาระบบสารสนเทศคือ การนิยามขอบเขตและสิ่งแวดล้อมที่เกี่ยวข้องกับระบบสารสนเทศให้ชัดเจนและครบถ้วนสมบูรณ์ที่สุด

3. ลักษณะของสารสนเทศที่ดี

ผู้ใช้งานแต่ละคนอาจต้องการรายละเอียดของสารสนเทศที่แตกต่างกัน แต่สิ่งหนึ่งที่ทุกคนต้องการเหมือนกันคือ ความมีคุณภาพของสารสนเทศนั้น ๆ ซึ่งโดยทั่วไปประกอบด้วยคุณลักษณะ ดังต่อไปนี้ (โอภาส เอี่ยมสิริวงศ์, 2558; ศรีไพร ศักดิ์รุ่งพงศากุล และเจษฎาพร ยุทธนวิบูลย์ชัย, 2549)

3.1 ความถูกต้องและเที่ยงตรง (Accuracy) สารสนเทศที่ดีจะต้องมีความเที่ยงตรงและมีความคลาดเคลื่อนน้อยที่สุด เพื่อให้การตัดสินใจในองค์กรถูกต้อง และสามารถวางแผนการปฏิบัติงานได้อย่างมีประสิทธิภาพ

3.2 ความสมบูรณ์ (Completeness) สารสนเทศที่ดีไม่ได้หมายความว่าเป็นสารสนเทศที่มีปริมาณมาก เท่านั้น เพราะถ้ามีมากเกินไป แต่รายละเอียดไม่ครบถ้วนและไม่ครอบคลุมทุกด้านก็ไม่สามารถใช้ประกอบการตัดสินใจได้

3.3 ความทันเวลาในการใช้งาน (Timelines) สารสนเทศที่ดีจะต้องสามารถเรียกใช้งานได้ทันทีที่ต้องการ นั่นก็หมายความว่าสารสนเทศจะต้องถูกเตรียมพร้อมสำหรับการใช้งานในองค์กรตลอดเวลา

3.4 ตรงกับความต้องการใช้งานของผู้ใช้ (Relevance) สารสนเทศที่ดีจะต้องตอบสนองต่อความต้องการใช้งานของทุกแผนกในองค์กร ดังนั้น การวิเคราะห์และออกแบบระบบสารสนเทศใด ๆ ในองค์กรจะต้องมีการรวบรวมความต้องการใช้งานจากผู้ที่เกี่ยวข้องทุกระดับ

3.5 ความทันสมัย (Up-to-date) สารสนเทศที่ดีจะต้องถูกปรับปรุง เปลี่ยนแปลงให้ทันสมัยอยู่ตลอดเวลา เพื่อให้ทันต่อสภาวการณ์ของทั้งภายในและภายนอกองค์กร

3.6 ตรวจสอบได้ (Verifiability) สารสนเทศที่ดีจะต้องตรวจสอบแหล่งที่มาของข้อมูลสารสนเทศได้ทั้งนี้เพื่อให้ความมั่นใจว่าสามารถนำมาประกอบการทำงานและการตัดสินใจได้ หรือว่าควรจะแสวงหาข้อมูล/สารสนเทศจากแหล่งที่มาอื่นที่น่าเชื่อถือมากกว่านี้

4. องค์ประกอบของระบบสารสนเทศ

การพัฒนาระบบสารสนเทศที่ใช้คอมพิวเตอร์เป็นพื้นฐานการทำงาน จำเป็นต้องตระหนักถึงองค์ประกอบหลักที่อยู่ภายใต้ขอบเขตความต้องการใช้งาน และสอดคล้องกับสิ่งแวดล้อมของระบบสารสนเทศนั้น ๆ (โอภาส เอี่ยมสิริวงศ์, 2558)

4.1 ฮาร์ดแวร์ (Hardware) เช่น เครื่องคอมพิวเตอร์แม่ข่าย เครื่องคอมพิวเตอร์ลูกข่าย เครื่องพิมพ์เครื่องอ่านบาร์โค้ด และอุปกรณ์ต่อพ่วงอื่น ๆ เป็นต้น

4.2 ซอฟต์แวร์ (Software) เช่น โปรแกรมที่ใช้พัฒนาระบบสารสนเทศ โปรแกรมในการจัดการฐานข้อมูล โปรแกรมสำหรับอุปกรณ์ต่อพ่วง และโปรแกรมพื้นฐานที่จำเป็นต้องใช้ประกอบการทำงาน เป็นต้น

4.3 เครือข่ายการสื่อสาร (Network/Communication) เช่น การเชื่อมโยงอุปกรณ์ต่าง ๆ เพื่อประหยัดทรัพยากรและการจัดการแลกเปลี่ยนข้อมูลร่วมกันระหว่างระบบสารสนเทศทั้งภายในและภายนอกองค์กร เป็นต้น

4.4 ผู้ใช้งาน (User) เช่น ผู้ใช้งานทุกระดับภายในองค์กรและภายนอกองค์กรที่มีหน้าที่ความรับผิดชอบเกี่ยวข้องกับการใช้งานระบบสารสนเทศนั้น ๆ

4.5 ข้อมูล (Data) เช่น ข้อมูลต่าง ๆ ที่จำเป็นต้องใช้เพื่อประมวลผลให้ได้สารสนเทศตามที่ผู้ใช้งานต้องการ

4.6 กฎระเบียบ (Regulation) เช่น กฎ ระเบียบ ข้อบังคับต่าง ๆ ทั้งจากภายในองค์กรและภายนอกองค์กรที่เกี่ยวข้องกับการพัฒนาระบบสารสนเทศนั้น ๆ ซึ่งจะเป็นหลักเกณฑ์ที่ใช้ในการพัฒนาระบบสารสนเทศ และทำให้นักพัฒนาระบบรู้ว่าจำเป็นต้องตรวจสอบความถูกต้องของข้อมูลที่จะนำเข้าสู่ระบบสารสนเทศอย่างไรบ้างเพื่อลดความผิดพลาดของข้อมูล และควรมีเงื่อนไขในการประมวลผลข้อมูลเป็นอย่างไร

4.7 เอกสาร/คู่มือ (Document/Manual) เช่น เอกสารที่เกี่ยวข้องกับการพัฒนาระบบสารสนเทศ โดยอาจเป็นเอกสารขั้นตอนการทำงานแบบเดิม เอกสารเชิงเทคนิคที่เกิดขึ้นในระหว่างการพัฒนาระบบสารสนเทศ รวมไปถึงคู่มือการใช้งาน/การติดตั้งระบบสารสนเทศและอุปกรณ์ต่อพ่วงต่าง ๆ เป็นต้น

5. ประเภทของระบบสารสนเทศ

ระบบสารสนเทศสามารถแบ่งได้เป็น 6 ประเภทหลัก ตามวัตถุประสงค์และคุณลักษณะของระบบ (อรยา ปรีชาพานิช, 2557) ดังนี้

5.1 ระบบประมวลสารสนเทศเชิงรายการ (Transaction Processing System : TPS)

ระบบประมวลสารสนเทศเชิงรายการ เป็นระบบสารสนเทศขั้นพื้นฐานของทุกองค์กรที่ใช้ในการจัดเก็บ ข้อมูลที่เกิดขึ้นจากการดำเนินงานประจำวันขององค์กรเข้าสู่ระบบสารสนเทศ เพื่อนำไปใช้ในการประมวลผลเป็นสารสนเทศต่อไป ยกตัวอย่างเช่น

1) การขายสินค้าที่ร้านสะดวกซื้อโดยใช้เครื่องอ่านบาร์โค้ดในการบันทึกรายการค้า

2) การยืม-คืนหนังสือของห้องสมุด

3) การบันทึกประวัติผู้ป่วยและผลการรักษาโรคในโรงพยาบาล

4) การลงทะเบียนเรียนของนิสิตผ่านเครือข่ายอินเทอร์เน็ต

คุณลักษณะสำคัญของ TPS ประกอบด้วย

1) การจัดเก็บ (Storage) เป็นการจัดเก็บข้อมูลและสารสนเทศจากการดำเนินงานประจำวันขององค์กร เพื่อนำไปใช้งานต่อไป โดยผู้ใช้สามารถเพิ่ม/แก้ไข/ลบข้อมูลให้ถูกต้องและทันสมัยตลอดเวลา

2) การจัดหมวดหมู่ (Classification) เป็นการจัดกลุ่มข้อมูลตามประเภทที่ต้องการใช้งาน เช่น การจัดหมวดหมู่สินค้าตามประเภทสินค้า การจัดหมวดหมู่ข้อมูลนักศึกษาแยกตามคณะและสาขาวิชา และการจัดรายวิชาแยกตามภาคการศึกษา เป็นต้น

3) การคำนวณ (Calculation) เป็นการคำนวณผลจากข้อมูลที่นำเข้าสู่ระบบ เช่น การคำนวณรายได้สุทธิในแต่ละวัน การคำนวณเกรดเฉลี่ยของนักศึกษาแต่ละคน การคำนวณค่าลงทะเบียนเรียนของนักศึกษาในแต่ละภาคการศึกษา เป็นต้น

4) การเรียงลำดับ (Sorting) เป็นการจัดเรียงลำดับข้อมูลและสารสนเทศเพื่อให้เกิดความสะดวกในการใช้งาน เช่น การแสดงข้อมูลใบสั่งซื้อสินค้าตามลำดับวันที่สั่งสินค้า การเรียงลำดับชื่อนักศึกษาตามรหัสนักศึกษา เป็นต้น

5) การสรุปผล (Summarization) เป็นการสรุปผลข้อมูลในระบบ เช่น การจัดพิมพ์ใบส่งของเรียงตามลำดับวันที่ส่งของ และการแสดงผลการเรียนของนักศึกษาแต่ละคนเรียงตามลำดับภาคการศึกษา เป็นต้น

5.2 ระบบสารสนเทศเพื่อการจัดการ (Management Information System : MIS)

ระบบสารสนเทศเพื่อการจัดการเป็นระบบที่ใช้ข้อมูลที่จัดเก็บไว้ใน TPS มาวิเคราะห์และสรุปผลใน รูปแบบรายงานเพื่อให้ผู้บริหารใช้ในการวางแผน การควบคุม และการแก้ปัญหาต่าง ๆ

คุณลักษณะสำคัญ MIS คือ เป็นระบบที่เน้นการจัดทำรายงานเพื่อการจัดการ โดยสามารถจำแนกรายงาน ออกเป็น 4 ประเภทหลักดังนี้

1) รายงานที่จัดทำขึ้นตามระยะเวลาที่กำหนด (Periodic Reports) โดยมีการกำหนดรูปแบบและ ระยะเวลาไว้ล่วงหน้า เช่น รายงานประจำวัน รายงานประจำสัปดาห์ และรายงานประจำเดือน เป็นต้น รายงานเหล่านี้ประกอบด้วยสารสนเทศที่มีประโยชน์ต่อการปฏิบัติงานของผู้บริหารชั้นต้นและผู้บริหาร ชั้นกลาง โดยอาจแสดงรายละเอียดหรือใช้ในรูปแบบสรุปก็ได้ขึ้นอยู่กับวัตถุประสงค์การใช้งาน

2) รายงานที่จัดทำตามความต้องการใช้งาน (Demand Reports/Ad-hoc Reports) มักเป็น

รายงานที่จัดทำขึ้นตามความต้องการใช้งานของผู้บริหารตามสถานการณ์ที่เกิดขึ้น ซึ่งรายงานเหล่านี้ จะถูกร้องขอเมื่อผู้บริหารไม่สามารถรอเวลาในการใช้งานตามระยะเวลาที่กำหนดได้ เช่น ผู้จัดการ ฝ่ายขายต้องการรายงานสถิติการขายสินค้าในระยะเวลา 6 เดือนที่ผ่านมา เพื่อนำมาวิเคราะห์และ กำหนดโปรโมชันส่งเสริมการขายอย่างเร่งด่วน เป็นต้น

3) รายงานตัวชี้วัดที่สำคัญ (Key Indicator Reports) เป็นรายงานสรุปกิจกรรมที่สำคัญขององค์กร เช่น ผลการผลิตสินค้า ยอดขายสินค้า และสถานะทางการเงิน เป็นต้น เพื่อช่วยให้ผู้บริหารสามารถบริหารจัดการองค์กรได้ตามเป้าหมายและแผนการดำเนินงานที่กำหนดไว้ล่วงหน้า

4) รายงานที่จัดทำตามเงื่อนไขเฉพาะ (Exception Reports) เพื่อใช้ประกอบการตัดสินใจแก้ปัญหาหนึ่ง ๆ ที่เกิดขึ้น ณ ขณะนั้น เช่น รายงานเกี่ยวกับปัญหาการปฏิบัติงานล่าช้า รายงานเกี่ยวกับลูกค้าที่ไม่ปฏิบัติตามสัญญา หรือรายงานสินค้าคงคลังที่ขาดหายไปจากโกดังสินค้า เป็นต้น

5.3 ระบบสารสนเทศเพื่อสนับสนุนการตัดสินใจ (Decision Support System : DSS)

ระบบสารสนเทศเพื่อสนับสนุนการตัดสินใจ เป็นระบบสารสนเทศที่นำข้อมูลจาก TPS และ MIS มาใช้งานรวมทั้งมีการนำข้อมูลจากภายนอกมาร่วมในการพิจารณาตัดสินใจและเสนอแนวทางให้แก่ผู้บริหาร คุณลักษณะสำคัญของ DSS จะมีส่วนประกอบและโครงสร้างของระบบ ดังนี้

1) การจัดการข้อมูล (Data Management) ประกอบด้วยฐานข้อมูลและวิธีการดึงข้อมูลที่เกี่ยวข้องกับสถานการณ์มาใช้งาน โดยอาจมาจากทั้งภายในและภายนอกองค์กร

2) การติดต่อระหว่างผู้ใช้และคอมพิวเตอร์ (User Interface) เพื่อช่วยให้ผู้ใช้สามารถสื่อสารและสั่งงานระบบ DSS ได้

3) การจัดการแบบจำลอง (Model Management) เพื่อใช้ในการวิเคราะห์ข้อมูลประกอบการตัดสินใจ DSS เป็นระบบที่มีความสามารถในการวิเคราะห์ในรูปแบบ Sensitivity Analysis ซึ่งเป็นการศึกษาถึงผลจากการเปลี่ยนแปลงของแบบจำลอง (Model) ส่วนใดส่วนหนึ่งหรือหลายส่วนว่า หากมีการเปลี่ยนแปลงปัจจัยนำเข้าแล้วจะส่งผลถึงผลลัพธ์อย่างไร โดยแบ่งได้เป็น 2 ประเภทหลักคือ

3.1) What-if Analysis เป็นการวิเคราะห์การเปลี่ยนแปลงปัจจัยนำเข้า (Input) ที่จะมีผลกระทบถึงผลลัพธ์(Output)

3.2) Goal Seeking Analysis เป็นการวิเคราะห์แบบถอยหลัง (Backward Solution) โดยวิเคราะห์หาปัจจัยนำเข้าที่เหมาะสมเพื่อให้ได้ผลลัพธ์ตามที่ได้กำหนดไว้

วัตถุประสงค์หลักของ DSS คือ เป็นเครื่องมือที่ช่วยให้ผู้บริหารมองเห็นทางเลือกที่เป็นไปได้ในการแก้ปัญหาหนึ่ง ๆ โดยผลลัพธ์ที่ได้จะเปลี่ยนแปลงไปตามตัวแปรต่าง ๆ ที่มีความเกี่ยวข้องกับปัญหานั้น ๆ จากนั้นผู้บริหารจึงตัดสินใจเลือกวิธีการแก้ปัญหาที่คิดว่าเหมาะสมที่สุดในสถานการณ์ดังกล่าว

5.4 ระบบสารสนเทศสำหรับผู้บริหาร (Executive Information System : EIS)

ระบบสารสนเทศสำหรับผู้บริหาร เป็นการนำสารสนเทศจากระบบ TPS MIS และ DSS มาใช้ร่วมกัน โดยอาศัยข้อมูลจากแหล่งข้อมูลภายในและภายนอกองค์กร ข้อมูลที่นำมาช่วยในการตัดสินใจจะคล้ายๆ กับ DSS แต่จะไม่เน้นตัดสินใจในระดับปฏิบัติการหรือแผนงานระยะสั้น นั่นคือ EIS จะเน้นการตัดสินใจจะเน้นในเชิงกลยุทธ์ (Strategy) สำหรับผู้บริหารระดับสูงที่มีความสามารถในการวิเคราะห์เปรียบเทียบและพยากรณ์แนวโน้มของสถานการณ์ โดยการใช้รูปภาพหรือแบบจำลองในการนำเสนอข้อมูล คุณลักษณะสำคัญของ EIS ประกอบด้วย

1) สนับสนุนการตัดสินใจเชิงกลยุทธ์ (Strategic Planning Support) ซึ่งถือว่าเป็นหน้าที่ความรับผิดชอบหลักของผู้บริหารระดับสูงขององค์กรที่มักมีโครงสร้างในการตัดสินใจแบบไม่มีโครงสร้าง

2) สามารถเชื่อมโยงกับสิ่งแวดล้อมภายนอกองค์กร (External Environment Focus) การวางแผน หรือตัดสินใจเชิงกลยุทธ์จำเป็นต้องใช้ข้อมูลและสารสนเทศจากภายนอก ดังนั้น ระบบ EIS จึงต้องเชื่อมโยงกับแหล่งข้อมูลเหล่านั้นเพื่อให้การดำเนินงานเป็นไปอย่างมีประสิทธิภาพ

3) สามารถคำนวณผลในภาพรวมได้ (Broad-Based Computing Capabilities) ผู้บริหารระดับสูงมักต้องการสารสนเทศที่มีความครอบคลุมในประเด็นต่าง ๆ ที่สนใจ โดยมีการนำเสนอในรูปแบบที่เข้าใจได้ง่าย เช่น ตารางหรือแผนภูมิในเชิงเปรียบเทียบด้านต่าง ๆ เป็นต้น

4) สามารถใช้งานได้ง่าย (Ease of Use) ผู้บริหารระดับสูงจะต้องการระบบสารสนเทศที่ใช้งานง่ายไม่ซับซ้อน เพื่อให้สามารถตอบสนองต่อความต้องการใช้งานสารสนเทศได้อย่างทันท่วงทีภายในสถานการณ์เร่งด่วน

5.5 ระบบสำนักงานอัตโนมัติ (Office Automation System : OAS)

ระบบสำนักงานอัตโนมัติ เป็นระบบการจัดการสารสนเทศในสำนักงานที่เกี่ยวข้องกับการรวบรวมข้อมูล การผลิตเอกสาร การสื่อสารข้อมูล และการสนับสนุนการปฏิบัติงานภายในองค์กร โดยนำเทคโนโลยีมาประยุกต์ใช้ เพื่อเพิ่มประสิทธิภาพในการทำงาน เช่น การจัดเตรียมข้อมูลและสารสนเทศสำหรับใช้ในการดำเนินงานการจัดตารางนัดหมาย การติดต่อประสานงานระหว่างแผนกต่างๆ ในองค์กร รวมไปถึงภายนอกองค์กร เป็นต้น

คุณลักษณะสำคัญของระบบสำนักงานอัตโนมัติสามารถแบ่งตามประเภทการใช้งานได้ดังนี้

1) ระบบจัดการด้านเอกสาร (Document Management System) เป็นระบบสำหรับจัดทำเอกสารต่าง ๆ ขององค์กรและจัดเก็บเอกสารที่เกี่ยวข้องกับองค์กรไว้อย่างเป็นหมวดหมู่

2) ระบบรับ-ส่งข่าวสาร (Message-handling System) เป็นระบบสำหรับรับ-ส่งข่าวสารต่าง ๆ ขององค์กร โดยการประยุกต์ใช้เทคโนโลยีสารสนเทศ เช่น ไปรษณีย์อิเล็กทรอนิกส์ (E-mail) และไปรษณีย์ เสียง (Voice Mail) เป็นต้น

3) ระบบประชุมทางไกล (Teleconferencing System) เป็นระบบสำหรับการประชุมร่วมกันจากต่างสถานที่เพื่อลดภาระการเดินทางของผู้เข้าร่วมประชุม โดยการประยุกต์ใช้เทคโนโลยีสารสนเทศ เช่น การประชุมทางไกลด้วยวิดีทัศน์ (Video Teleconferencing) และการประชุมทางไกลด้วยเสียง (Audio Teleconferencing) เป็นต้น

4) ระบบสนับสนุนงานสำนักงาน (Office Support System) เป็นระบบสำหรับสนับสนุนการปฏิบัติงานของพนักงานในองค์กร โดยการประยุกต์ใช้เทคโนโลยีสารสนเทศ เช่น การใช้กระดานข่าวอิเล็กทรอนิกส์ (Electronic Bulletin Board) การจัดการตารางนัดหมาย (Calendar Management) และการใช้กรุ๊ปแวร์ (Groupware) เป็นต้น

โดยประโยชน์ของระบบสำนักงานอัตโนมัติ สามารถสรุปเป็นประเด็นหลักได้ดังนี้

1) ประหยัดค่าใช้จ่าย โดยเฉพาะการจัดเตรียมเอกสารกระดาษ การจัดส่ง การรับ การจัดเก็บและการทำลาย รวมทั้งงบประมาณในการจัดจ้างผู้ดำเนินการในแต่ละขั้นตอน

2) เพิ่มประสิทธิภาพในสำนักงาน ลดขั้นตอนการทำงานและลดโอกาสในการพิมพ์ผิดพลาด

3) ผู้บริหารสามารถตัดสินใจได้ถูกต้องรวดเร็วขึ้น เนื่องจากความถูกต้อง แม่นยำ และความรวดเร็วในการสืบค้นข้อมูล

4) บุคลากรมีความภาคภูมิใจในองค์กรมากขึ้น เนื่องจากมีสำนักงาน และเครื่องมืออุปกรณ์ที่ทันสมัย ทำให้ประหยัดเวลาในการทำงาน

5) องค์กรมีภาพลักษณ์ที่ดีสำหรับหน่วยงานภายนอก ที่ได้รับการบริการและการติดต่อสื่อสารที่ถูกต้องรวดเร็วทันสมัย

5.6 ระบบจัดการความรู้ (Knowledge Management System : KMS)

ระบบจัดการความรู้เป็นระบบที่รวบรวมความรู้ที่มีอยู่กระจัดกระจายในองค์กรมาจัดไว้เป็นหมวดหมู่ เพื่อเป็นทรัพยากรหลักประกอบการทำงาน และบำรุงรักษาให้มีความทันสมัยอยู่เสมอ โดยจัดช่องทางการเข้าถึงความรู้ให้สะดวก รวดเร็ว และทั่วถึง เพื่อให้บุคลากรนำความรู้ไปพัฒนาการปฏิบัติงานให้มีประสิทธิภาพสูงสุด

องค์กรในปัจจุบันได้ให้ความสำคัญกับการเป็นองค์กรแห่งการเรียนรู้ (Learning Organizations) เพื่อ ความอยู่รอดและมั่นคงภายใต้สภาพแวดล้อมทางธุรกิจที่มีการเปลี่ยนแปลงอย่างรวดเร็ว จึงจำเป็นต้องมีการสร้างความรู้ทางธุรกิจใหม่ ๆ และเผยแพร่ความรู้ภายในองค์กรเพื่อสร้างหรือปรับปรุงผลิตภัณฑ์และบริการให้มีคุณภาพ มากยิ่งขึ้น รวมทั้งหาวิธีการหรือเทคนิคในการจัดการความรู้และเทคโนโลยีสารสนเทศเพื่อส่งเสริมให้พนักงานมีส่วนร่วมในการสะสมความรู้เหล่านั้น

ระบบจัดการความรู้จะช่วยให้พนักงานที่เป็นผู้รู้ได้สร้างและจัดระเบียบ รวมทั้งแบ่งปันความรู้ได้อย่างไม่จำกัดสถานที่และระยะเวลา ตัวอย่างเช่น ระบบจัดการความรู้ที่ใช้ระบบเครือข่ายอินเทอร์เน็ต

อินทราเน็ต และเว็บบอร์ด เป็นเทคโนโลยีหลักสำหรับการรวบรวม จัดเก็บและเผยแพร่ความรู้ทางธุรกิจ เพื่ออำนวยความสะดวก ในการเรียนรู้ร่วมกันของบุคลากรในองค์กร

ความหมายของนวัตกรรม

พจนานุกรมฉบับราชบัณฑิตยสถาน พ.ศ. 2557 (ม.ป.ป.) ได้ให้คำนิยามคำว่า นวัตกรรม ไว้ว่า “การกระทำหรือสิ่งที่ทำขึ้นใหม่หรือแปลกจากเดิมซึ่งอาจจะเป็นความคิด วิธีการ หรืออุปกรณ์ เป็นต้น”

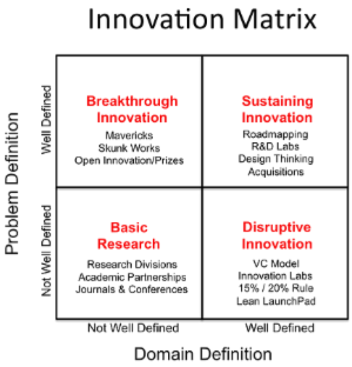
สำนักนวัตกรรมแห่งชาติ (2562) ได้ให้คำนิยามคำว่า นวัตกรรม ไว้ว่า “สิ่งใหม่ที่เกิดจากการใช้ความรู้และความคิดสร้างสรรค์ที่มีประโยชน์ต่อเศรษฐกิจและสังคม”

สมนึก เอื้อจิระพงษ์พันธ์ และคณะ (2553) ได้ให้ความหมายของ นวัตกรรม หมายถึง “สิ่งใหม่ที่เกิดขึ้นจากการใช้ความรู้ทักษะ ประสบการณ์และความคิดสร้างสรรค์ในการพัฒนาขึ้น ซึ่งอาจจะมีลักษณะเป็นผลิตภัณฑ์ใหม่ บริการใหม่ หรือกระบวนการใหม่ ที่ก่อให้เกิดประโยชน์ในเชิงเศรษฐกิจและสังคม”

ดังนั้นจึงสามารถสรุปได้ว่า นวัตกรรม (Innovation) หมายถึง กระบวนการหรือผลิตภัณฑ์ใหม่ที่สร้างขึ้นโดยมีการนำเสนอแนวคิดใหม่ หรือการปรับเปลี่ยนแนวคิดที่มีอยู่ให้มีคุณค่าและสร้างสรรค์ หรือการสร้างผลิตภัณฑ์หรือบริการใหม่เพื่อตอบสนองความต้องการของตลาดหรือกลุ่มลูกค้าใหม่

1. ประเภทของนวัตกรรม

นวัตกรรม (Innovation) สามารถอธิบายตาม Innovation Matrix ซึ่งเป็นแผนภาพหรือแมทริกซ์ที่ใช้เพื่ออธิบายและจัดเรียงความสำคัญของสิ่งต่าง ๆ ที่เกี่ยวข้องกับนวัตกรรม โดยเฉพาะเมื่อเปรียบเทียบนวัตกรรมใหม่ที่สามารถเป็นผู้นำในตลาดหรือเป็นสิ่งที่สำคัญในการพัฒนาธุรกิจ การใช้ Innovation Matrix ช่วยให้องค์กรหรือผู้บริหารสามารถประเมินและวางแผนนวัตกรรมอย่างมีระบบและมีข้อมูลมากขึ้น ดังภาพที่



ภาพที่ 1.1 อธิบายประเภทนวัตกรรมด้วย Innovation Matrix

ที่มา : Satell, G., (2017)

จากภาพที่ 1.1 Innovation Matrix (Satell, G., 2017) เป็นแนวคิดที่ใช้เพื่อคำนวณและแบ่งประเภทนวัตกรรมหรือโครงการนวัตกรรมตามการระบุปัญหาและการระบุขอบเขต (Problem Definitionและ Domain Definition) อย่างชัดเจนหรือไม่ ทั้งสองมุมมองนี้มีบทบาทสำคัญในการกำหนดความเป็นไปได้และความคาดหวังของโครงการนวัตกรรม ได้แก่ 1) Problem Definition (การระบุปัญหา) ความชัดเจนในการระบุปัญหาคือการระบุปัญหาหรือความต้องการที่ต้องการแก้ได้อย่างชัดเจน ปัญหาที่ระบุได้ชัดเจนมีลักษณะที่เน้นเรื่องหรือความจำเป็นของการแก้ไขปัญหานั้น ๆ ในบริบทของลูกค้าหรือผู้ใช้บริการ 2) Domain Definition (การระบุขอบเขต) เน้นการระบุวิธีการแก้ไขปัญหา และผู้แก้ไขปัญหาได้ชัดเจน นั่นหมายถึงการกำหนดว่าคือใครหรือองค์กรใดที่จะรับผิดชอบในการแก้ไขปัญหานั้น และวิธีการที่จะใช้ในการแก้ไขปัญหานั้นให้เป็นไปได้อย่างชัดเจน

สามารถที่จะอธิบายประเภทของนวัตกรรมตาม Innovation Matrix แบ่งเป็น 4 ประเภท (Satell, G., 2017) ดังนี้

1.1 วิจัยพื้นฐาน (Basic research)

งานวิจัยที่เน้นการสร้างความรู้และความเข้าใจเบื้องต้นในแวดล้อมที่ไม่แปรผัน มักไม่มีการนำความรู้ดังกล่าวมาใช้ในสิ่งต่อไปในทันที แต่เป็นการสร้างพื้นฐานความรู้สำคัญสำหรับวิทยาศาสตร์และเทคโนโลยีในอนาคต ตัวอย่างเช่น งานวิจัยในด้านวิทยาศาสตร์พื้นฐานเพื่อทราบความก้าวหน้าในฟิสิกส์หรือเคมี ซึ่งอาจจะไม่มีการนำไปใช้ในสิ่งต่อไปในทันทีแต่มีผลทางวิทยาศาสตร์

1.2 นวัตกรรมแบบก้าวล้ำ (Breakthrough innovation)

เหมาะกับการแก้ปัญหาที่ยาก ๆ เรื้อรัง ที่แก้ไม่ได้ซักที ปัญหาเหล่านี้จะแก้ได้จากการคิดนอกกรอบ และมีกลยุทธ์ในการแก้ปัญหาเหล่านี้ จึงเหมาะกับการสร้างทีมใหม่ ๆ ที่ให้ผู้คนหลากหลายสาขาวิชามาแก้ปัญหาร่วมกัน หรือเปิดโจทย์ประกวดให้คนนอกมาช่วยแก้ปัญหาให้

1.3 นวัตกรรมแบบยั่งยืน (Sustaining Innovation)

นวัตกรรมแบบยั่งยืน คือ นวัตกรรมที่เกิดขึ้นจากเทคโนโลยีที่ไม่ได้ใหม่มาก แต่มีผลกระทบสูงต่อผู้ใช้งาน มักใช้ทำให้ตลาดเติบโตขึ้น เพื่อกอบโกยกำไรโดยไม่มีการหาตลาดใหม่ โดยมากมักเป็นนวัตกรรมที่ค่อย ๆ พัฒนาไปทีละเล็กทีละน้อย แต่จะเป็นนวัตกรรมที่ดีกว่า เร็วกว่า ถูกกว่า เน้นความต้องการของผู้ใช้งานและธุรกิจมากกว่าเทคโนโลยี

1.4 นวัตกรรมแบบพลิกโฉม (Disruptive Innovation)

นวัตกรรมแบบพลิกโฉม คือ นวัตกรรมที่เกิดขึ้นจากเทคโนโลยีใหม่และมีผลกระทบสูงต่อผู้ใช้งาน นิยมใช้สร้างตลาดใหม่อย่างที่ไม่เคยมีมาก่อน เป็นนวัตกรรมที่เกิดขึ้นเพื่อทดแทนของเก่าโดยสิ้นเชิง ตัวอย่างการนวัตกรรมแบบนี้เช่น การเข้ามาของ Google Maps ทำให้ผู้คนหันมาใช้แทนแผนที่แบบกระดาษ

การใช้แอปพลิเคชันในการเรียกแท็กซี่แทนการโบก เพื่อลดการปฏิเสธการเดินทาง การเข้ามาของ Netflix ซึ่งทดแทนร้านเช่าแผ่นภาพยนตร์

2. รูปแบบนวัตกรรม

รูปแบบของนวัตกรรมออกเป็น 3 กลุ่มใหญ่ ๆ ตามลักษณะของการเพิ่มคุณค่าในระดับต่าง ๆ ดังนี้

2.1 นวัตกรรมผลิตภัณฑ์ (Product Innovation)

นวัตกรรมประเภทนี้เกี่ยวข้องกับการสร้างผลิตภัณฑ์ใหม่หรือพัฒนาผลิตภัณฑ์ที่มีอยู่ให้ดียิ่งขึ้นกว่าเดิม โดยการเพิ่มคุณสมบัติ คุณภาพ หรือความสามารถของสินค้าหรือบริการ เช่น การปรับเปลี่ยนดีไซน์สินค้า การนำเอาวัตถุดิบใหม่เข้ามาใช้ หรือการเปลี่ยนแปลงบรรจุภัณฑ์สินค้าเพื่อเพิ่มความน่าสนใจให้กับลูกค้า

2.2 นวัตกรรมการดำเนินงาน (Process Innovation)

ประเภทนี้เกี่ยวข้องกับการพัฒนากระบวนการดำเนินงานหรือวิธีการทำงานภายในองค์กรเพื่อเพิ่มประสิทธิภาพ ลดต้นทุน หรือเพิ่มคุณภาพของผลิตภัณฑ์หรือบริการ เช่น การเปลี่ยนแปลงวิธีการผลิตเพื่อลดระยะเวลาและการสูญเสีย การปรับปรุงกระบวนการบริหารจัดการ หรือการใช้เทคโนโลยีในกระบวนการผลิตเพื่อเพิ่มประสิทธิภาพ

2.3 นวัตกรรมธุรกิจ (Business Innovation)

ประเภทนี้เกี่ยวข้องกับการเปลี่ยนแปลงแนวทางทางธุรกิจหรือกลยุทธ์ขององค์กร เพื่อเพิ่มมูลค่าให้กับลูกค้าและตลาด เช่น การพัฒนาบริการหรือผลิตภัณฑ์ใหม่เพื่อตอบโจทย์ความต้องการของตลาด การเปลี่ยนแปลงโมเดลธุรกิจเพื่อเข้าสู่ตลาดใหม่ หรือการใช้เทคโนโลยีในการแก้ไขปัญหาในการดำเนินธุรกิจ

การแบ่งประเภทนวัตกรรมเป็นรูปแบบต่าง ๆ ช่วยให้เราเข้าใจลักษณะและความเป็นมาของนวัตกรรมในแต่ละด้านได้มากขึ้น และช่วยในการวางแผนและดำเนินการสู่การพัฒนาและการเติบโตขององค์กรอย่างมีประสิทธิภาพ

ความหมายของเทคโนโลยี

คำว่า “เทคโนโลยี” (Technology) เกิดมาจากคำภาษากรีก “Technologia” ซึ่งมีความหมายว่า "การกระทำอย่างมีระบบ" หมายความว่าเป็นกระบวนการที่นำเอาความรู้ทางวิทยาศาสตร์ แนวความคิด ประสบการณ์ วิธีการที่เป็นระบบและระเบียบ รวมถึงผลิตผลด้านวิทยาศาสตร์และวิศวกรรม เพื่อนำมาประยุกต์ใช้ให้เกิดประโยชน์ในทางปฏิบัติและอุตสาหกรรม นอกจากนี้ พจนานุกรมไทยฉบับราชบัณฑิตยสถาน พ.ศ. 2542 (2546) ได้อธิบายคำว่า “เทคโนโลยี” ว่าเป็นวิทยาการที่เกี่ยวกับการนำเอาวิทยาศาสตร์มาประยุกต์ใช้ให้เกิดประโยชน์ในทางปฏิบัติและอุตสาหกรรมหรือเป็นการประยุกต์ความรู้ด้านวิทยาศาสตร์ให้ใช้เพื่อแก้ไขปัญหาต่าง ๆ และสร้างสรรค์วัสดุอุปกรณ์ เครื่องมือ และเครื่องจักร รวมถึงการสร้างความรู้ใหม่ ๆ เพื่อให้การดำรงชีวิตของมนุษย์เป็นไปอย่างง่ายและสะดวกยิ่งขึ้น

1. ลักษณะของเทคโนโลยี

เทคโนโลยีมีลักษณะที่สามารถจำแนกได้เป็น 3 ลักษณะหลักดังนี้

1.1 เทคโนโลยีในลักษณะของกระบวนการ (Process Technology)

เป็นการนำเอาความรู้และวิธีการทางวิทยาศาสตร์มาใช้เพื่อดำเนินกระบวนการต่าง ๆ อย่างมีระบบและเป็นระเบียบ เช่น กระบวนการผลิตที่มีการปรับปรุงเพื่อเพิ่มประสิทธิภาพ หรือวิธีการการจัดการที่มีการพัฒนาเพื่อให้เกิดผลลัพธ์ที่ดีขึ้น

1.2 เทคโนโลยีในลักษณะของผลผลิต (Product Technology)

เป็นการพัฒนาวัสดุหรือผลิตภัณฑ์ที่มีการปรับปรุงและพัฒนาขึ้นใหม่ หรือการสร้างผลิตภัณฑ์ใหม่ที่มีคุณสมบัติหรือความสามารถที่แตกต่างจากเดิม เช่น การพัฒนาเทคโนโลยีในอุตสาหกรรมเครื่องใช้ไฟฟ้า หรือการสร้างผลิตภัณฑ์ใหม่ในตลาดเทคโนโลยี

1.3 เทคโนโลยีในลักษณะผสมของกระบวนการและผลผลิต (Process and Product Technology)

เป็นการรวมการพัฒนากระบวนการและผลผลิตเข้าด้วยกัน โดยเน้นที่การปรับปรุงกระบวนการเพื่อเพิ่มคุณค่าให้กับผลผลิต เช่น ระบบคอมพิวเตอร์ที่มีการปรับปรุงทั้งกระบวนการทำงานและคุณสมบัติของผลิตภัณฑ์

2. องค์ประกอบของเทคโนโลยี

ระบบเทคโนโลยีถูกสร้างขึ้นโดยมนุษย์เพื่อแก้ไขปัญหาและตอบสนองความต้องการ ระบบเทคโนโลยีประกอบด้วยองค์ประกอบหลัก 5 ส่วนหลัก ได้แก่

2.1 ปัจจัยนำเข้า (Input) หมายถึง ปัญหาที่ต้องการแก้ไขหรือความต้องการของมนุษย์ที่เราต้องการตอบสนอง

2.2 กระบวนเทคโนโลยี (Process Technology) คือ ขั้นตอนแก้ไขปัญหาหรือตอบสนองความต้องการของมนุษย์ โดยใช้ทรัพยากรทางเทคโนโลยีเพื่อให้ได้ผลผลิตตามความต้องการ

2.3 ผลผลิต (Output) เป็นผลสุดท้ายที่ได้มาจากการใช้เทคโนโลยี ซึ่งอาจเป็นชิ้นงานหรือวิธีการตอบสนองความต้องการ

2.4 ทรัพยากรทางเทคโนโลยี (Technology Resources) เป็นสิ่งจำเป็นที่ใช้ในการแก้ปัญหาให้ได้ผลลัพธ์ตามความต้องการ เช่น มนุษย์ ข้อมูลสารสนเทศ เครื่องมือ พลังงานทุนทรัพย์สิน และเวลาเป็นต้น

2.5 ปัจจัยที่มีส่วนขัดขวาง (Constraintsคือ ปัจจัยที่เข้ามาเป็นข้อจำกัดหรือข้อพิจารณาที่ต้องพิจารณาในกระบวนการแก้ปัญหา เช่น ข้อจำกัดในทรัพยากร หรือเวลาที่จำกัด เป็นต้น

การนำเทคโนโลยีมาใช้ถูกต้องและเหมาะสมจะส่งผลให้เกิดการพัฒนาที่เป็นประโยชน์ต่อการดำรงชีวิตของทุกคน ปัจจุบันเทคโนโลยีมีบทบาทสำคัญในการดำรงชีวิต เนื่องจากมีความเกี่ยวข้องกับวิทยาศาสตร์และเทคโนโลยีตลอดเวลา

ความหมายของธุรกิจสมัยใหม่

กระแสสำคัญของธุรกิจในปัจจุบันเริ่มเปลี่ยนแปลงอย่างรวดเร็วด้วยการเคลื่อนไหวของโลกออนไลน์ ในยุคที่เทคโนโลยีและการเชื่อมต่ออินเทอร์เน็ตเข้าสู่ทุกด้านของชีวิต สื่อออนไลน์เข้ามามีบทบาทที่มีน้ำหนักอย่างสำคัญในการเปลี่ยนแปลงพฤติกรรมของผู้บริโภคและกิจการ การเข้าสู่ยุคดิจิทัลนำพาการเกิดภาวะการแข่งขันที่เข้มข้นขึ้น การใช้เทคโนโลยีและสื่อออนไลน์เป็นเครื่องมือสำคัญในการต่อสู้เพื่อสร้างความแตกต่างและเพิ่มประสิทธิภาพในกิจการในหลากหลายอุตสาหกรรม การเข้าใจและนำการปรับตัวตามกระแสดังกล่าวเป็นสิ่งสำคัญที่ผู้ประกอบการต้องพิจารณาในการจัดการธุรกิจของตน

การเปลี่ยนแปลงในพฤติกรรมของผู้บริโภคและที่ตั้งของกิจการถูกผลักดันอย่างมากโดยการเคลื่อนไหวของสื่อออนไลน์ นั่นคือผู้บริโภคมีความสามารถในการเข้าถึงข้อมูลและสินค้าได้ง่ายๆ และตัดสินใจซื้อสินค้าหรือใช้บริการได้ทันทีผ่านทางอินเทอร์เน็ต ธุรกิจที่เข้าใจและนำเอากระแสดังกล่าวมาเป็นส่วนหนึ่งของกลยุทธ์การตลาดจะมีโอกาสเติบโตอย่างรวดเร็วกว่าคู่แข่ง

ในฐานะที่สื่อออนไลน์กลายเป็นช่องทางที่สำคัญในการสื่อสารกับกลุ่มลูกค้า นักการตลาดต้องวางแผนและดำเนินกิจกรรมการตลาดให้เข้ากับสื่อและโพรโมตที่เกี่ยวข้องกับออนไลน์ เช่น การใช้ Search Engine Optimization (SEO) เพื่อเพิ่มโอกาสให้สินค้าหรือบริการขององค์กรปรากฏในผลการค้นหาและเพิ่มโอกาสที่จะถูกคลิกเข้าชม การทำ Email Marketing เพื่อส่งข้อมูลและโปรโมชั่นถึงกลุ่มลูกค้าที่ต้องการ และการนำ Social Network เป็นช่องทางในการสร้างความสัมพันธ์และการติดต่อสื่อสารกับลูกค้าโดยตรง ความสำคัญของการทำการตลาดดิจิทัลยังเน้นให้สินค้าและบริการสามารถเข้าถึงลูกค้าได้อย่างรวดเร็วทั้งในสถานที่และเวลา โดยที่ผู้ประกอบการสามารถตอบสนองความต้องการของลูกค้าอย่างทันที และสร้างประสบการณ์การสื่อสารที่สร้างความเชื่อมั่นและความสัมพันธ์ใกล้ชิดขึ้น

การทำการตลาดดิจิทัลไม่เพียงแค่การโฆษณาผ่านช่องทางออนไลน์เท่านั้น แต่ยังเกี่ยวข้องกับการทำธุรกิจทั้งในด้านการวางแผน การวิเคราะห์ผู้บริโภค การจัดการสื่อสาร และการสร้างความสัมพันธ์ระยะยาวกับลูกค้า การทำการตลาดดิจิทัลมีคุณสมบัติพิเศษที่ช่วยลดค่าใช้จ่ายในการตลาดและเพิ่มประสิทธิภาพในการวางแผนและดำเนินกิจกรรม ซึ่งจะสนับสนุนให้ธุรกิจสามารถปรับตัวตามการเปลี่ยนแปลงของกลุ่มลูกค้าและสภาวะการแข่งขันได้อย่างมีประสิทธิภาพและได้ผลอย่างยั่งยืนในยุคดิจิทัลปัจจุบัน (นฤมล ญาณสมบัติ และเมธารัตน์ จันตะนี, 2562) ในหัวข้อนี้มีเนื้อหาเกี่ยวกับธุรกิจสมัยใหม่ ดังต่อไปนี้

1. ธุรกิจยุคดิจิทัล

ธุรกิจดิจิทัลหรือธุรกิจอิเล็กทรอนิกส์ (Electronic Business หรือ E-business) คือ กิจกรรมทางธุรกิจที่ดำเนินการผ่านระบบออนไลน์หรือระบบอินเทอร์เน็ตซึ่งกระบวนการหลักในการดำเนินธุรกิจแบบดิจิทัลนั้นครอบคลุมตั้งแต่ต้นทางที่เกี่ยวข้องกับการเชื่อมโยงกับซัพพลายเออร์ การทำงานต่าง ๆ และปลายทางที่เชื่อมโยงกับลูกค้าผ่านแพลตฟอร์มออนไลน์

ธุรกิจดิจิทัลนั้นประกอบด้วยหลายองค์ประกอบ เช่น การค้าขายออนไลน์ (E-commerce) ที่เกี่ยวข้องกับกระบวนการซื้อขายผ่านระบบอินเทอร์เน็ต เช่นการซื้อขายสินค้าผ่านเว็บไซต์ทั้งโดยตรงและผ่านแพลตฟอร์มต่าง ๆ อย่างเช่น Lazada, Shopee, Amazon, Ebay, Alibaba และอื่น ๆ และยังรวมถึงการบริการลูกค้าผ่านช่องทางออนไลน์ เช่น การให้บริการทางอินเทอร์เน็ต การติดต่อลูกค้าผ่านโซเชียลมีเดีย และแพลตฟอร์มต่าง ๆ ที่ให้บริการแก่ลูกค้า

นอกจากนี้ ธุรกิจดิจิทัลยังรวมถึงการใช้เทคโนโลยีและแพลตฟอร์มทางดิจิทัลเพื่อการเรียนรู้และการพัฒนาทักษะ เช่น แพลตฟอร์มการเรียนออนไลน์เช่น Coursera และ Udemy ซึ่งเปิดโอกาสให้ผู้คนสามารถเรียนรู้ใหม่ๆ และพัฒนาความรู้ทักษะได้ผ่านช่องทางออนไลน์ ทั้งนี้ ธุรกิจดิจิทัลไม่เพียงเพียงแค่การซื้อขายออนไลน์เท่านั้น แต่ยังเกี่ยวข้องกับการสร้างความเชื่อมั่นให้กับลูกค้าผ่านแพลตฟอร์มดิจิทัล การบริหารจัดการทรัพยากรที่มีอยู่ให้เป็นประสิทธิภาพ เพื่อสร้างความสำเร็จและเติบโตในธุรกิจออนไลน์อย่างยั่งยืน

2. ลักษณะของธุรกิจดิจิทัล

ธุรกิจดิจิทัล (Digital Business) มีลักษณะพิเศษที่แตกต่างจากธุรกิจทั่วไป โดยมีลักษณะ (ลั่นทม จอนจวบทรง และณธภร ธรรมบุญวริศ, 2562) ดังนี้

2.1 ทรัพยากรทางกายภาพมีจำนวนน้อย

ธุรกิจดิจิทัลมักมีความจำเป็นในการใช้ทรัพยากรทางกายภาพอย่างน้อยลง เนื่องจากส่วนใหญ่ของกิจกรรมจะเกิดขึ้นออนไลน์ ทำให้ไม่จำเป็นต้องมีพื้นที่สำหรับการจัดเก็บสินค้าหรือการให้บริการแบบดั้งเดิมเท่ากับธุรกิจแบบทั่วไป

2.2 เน้นความสร้างสรรค์

ธุรกิจดิจิทัลมีการเน้นการสร้างสรรค์และนวัตกรรม เนื่องจากสามารถใช้เทคโนโลยีและข้อมูลเพื่อพัฒนาผลิตภัณฑ์หรือบริการใหม่ ๆ อย่างรวดเร็ว และตอบสนองความต้องการของลูกค้าอย่างเชี่ยวชาญ

2.3 ใช้เครื่องมือหรือเทคโนโลยีเพื่อสร้างความชาญฉลาด

ธุรกิจดิจิทัลใช้เครื่องมือและเทคโนโลยีเพื่อทำงานอย่างเชี่ยวชาญ ในการวิเคราะห์ข้อมูลเข้าถึงข้อมูล และสร้างกลยุทธ์ต่างๆ เพื่อสร้างมูลค่าแก่ธุรกิจ

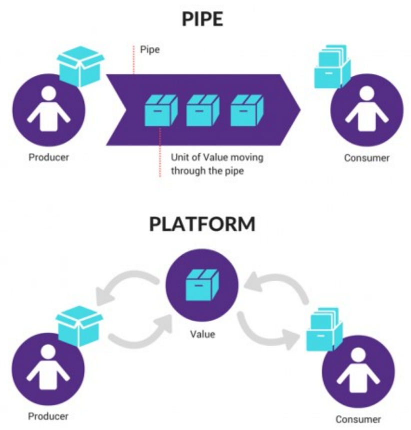
2.4 มีความรวดเร็วในบริการและการปรับตัว

ธุรกิจดิจิทัลมีความยืดหยุ่นและรวดเร็วในการปรับตัวต่อการเปลี่ยนแปลงของตลาด สามารถปรับแผนการตลาดหรือการขายสินค้าได้อย่างรวดเร็วเพื่อตอบสนองความต้องการของลูกค้าในขณะที่เกิดเหตุการณ์

ธุรกิจดิจิทัลนั้นเป็นแนวทางการพัฒนาธุรกิจที่ใช้เทคโนโลยีและอินเทอร์เน็ตเป็นเครื่องมือสำคัญเพื่อสร้างความนวัตกรรม ประสิทธิภาพและความรวดเร็วในการเติบโตและเป็นที่ยอมรับอย่างกว้างขวางในยุคดิจิทัลที่เปลี่ยนแปลงอย่างรวดเร็ว

3. รูปแบบธุรกิจดิจิทัล

รูปแบบธุรกิจดิจิทัลมุ่งเน้นการแบ่งปันมูลค่าในรูปแบบเศรษฐกิจแบ่งปัน (Shared Economy) และการดำเนินกิจกรรมในลักษณะปัจจุบันหรือเรียลไทม์บนแพลตฟอร์ม (Live or Real-time Market) โดยธุรกิจดิจิทัลจะใช้แพลตฟอร์ม (Platform) เป็นเครื่องมือหลักในการสร้างและแบ่งปันมูลค่า แพลตฟอร์มเหล่านี้จะต้องมีลักษณะสำคัญที่สร้างความเชื่อมโยงระหว่างผู้ใช้งาน โดยครอบคลุมผู้ผลิตหรือผู้จัดจำหน่าย ผู้ให้บริการ และลูกค้า เพื่อสร้างมูลค่าและการเชื่อมโยงในขณะที่กิจกรรมกำลังเกิดขึ้น ธุรกิจดิจิทัลจึงมุ่งหาแพลตฟอร์มหรือพัฒนาแพลตฟอร์มเพื่อเป็นศูนย์กลางในการสร้างความมั่งคั่ง โดยการสร้างความประทับใจในการเข้าร่วมใช้แพลตฟอร์ม และเพื่อให้ผู้ผลิต ผู้จัดจำหน่าย ผู้ให้บริการ และลูกค้าสามารถร่วมกันสร้างมูลค่าได้อย่างเต็มที่ โดยแบ่งปันเป็นหน่วยสร้างมูลค่าที่สามารถดึงดูดผู้ใช้งานได้ทั้งฝ่ายผู้ผลิต ผู้จัดจำหน่าย ผู้ให้บริการและลูกค้า เพื่อสร้างความเชื่อมโยงและมูลค่าในระบบแพลตฟอร์มนั้น รูปแบบธุรกิจดิจิทัลมีหลายแบบเนื่องจากมีลักษณะและลักษณะเฉพาะของแต่ละธุรกิจ (Parker, Van Alstyne และ Choudary, 2016) ดังแสดงในภาพที่ 1.2 ดังนี้



ภาพที่ 1.2 ความแตกต่างระหว่างธุรกิจรูปแบบ PIPE กับรูปแบบ PLATFORM
ที่มา : (Parker, Van Alstyne และ Choudary, 2016)

จากภาพที่ 1.2 อธิบายถึงแนวคิดที่เกี่ยวข้องกับแพลตฟอร์มและการเปลี่ยนแปลงในเศรษฐกิจดิจิทัล รูปแบบธุรกิจแบบ PIPEหรือ แบบท่อน้ำ หมายถึงแนวคิดเดิม ๆ ของธุรกิจที่เน้นการผลิตสินค้าหรือบริการเป็นหลัก รายได้ในรูปแบบ PIPE มักเกิดจากกระบวนการผลิตและขายสินค้าหรือบริการ โดยมุ่งหมายที่จะผลิตสินค้าหรือบริการที่มีคุณค่าสำหรับลูกค้าและตลาด แต่ธุรกิจรูปแบบ PLATFORM เป็นแนวคิดทางธุรกิจที่เน้นการสร้างแพลตฟอร์ม (Platform) เพื่อเชื่อมโยงระหว่างผู้ใช้และผู้ให้บริการ โดยอนุญาตให้พวกเขาแลกเปลี่ยนข้อมูล บริการ สินค้า หรือความรู้ รูปแบบธุรกิจแบบแพลตฟอร์ม มักมีลักษณะต่อไปนี้

3.1 โบรกเกอร์โมเดล (Brokerage model)

ธุรกิจที่เป็นนายหน้าในการเชื่อมโยงลูกค้ากับผู้ผลิตหรือผู้จัดจำหน่าย และผู้ให้บริการ เพื่อให้เกิดการซื้อขายหรือการใช้บริการ ซึ่งแพลตฟอร์มในโบรกเกอร์โมเดลสามารถเป็นตลาดเสมือนจริงออนไลน์ เช่น Lazada, Shopee, Tmall, Taobao ที่เป็นตลาดออนไลน์สำหรับการซื้อขายสินค้า หรือเป็นนายหน้าประมูลออนไลน์เช่น eBay.com ที่เปิดโอกาสให้ผู้ซื้อประมูลสินค้า และยังสามารถเป็นศูนย์รวมความต้องการสินค้าและบริการอย่าง Agoda.com หรือ Airbnb.com ที่เป็นแพลตฟอร์มให้ผู้ใช้บริการจองที่พัก นอกจากนี้ยังมีรูปแบบอื่น ๆ เช่น นายหน้าทำรายการทางธุรกิจ เช่น PayPal, Visa card หรือ Alipay ที่เป็นตัวแทนการชำระเงินในการดำเนินกิจกรรมทางธุรกิจ หรือตัวแทนกระจายสินค้า เช่น Kerry, DHL, UPS ที่ให้บริการจัดส่งสินค้าให้ผู้จัดจำหน่ายออนไลน์ และยังมีรูปแบบอื่น ๆ ในลักษณะ ให้บริการเรียกใช้บริการ (Hailing model) เช่น Line Man และ Grab ที่เป็นคนกลางที่เชื่อมโยงระหว่างผู้โดยสารหรือผู้ใช้บริการกับบริการรถแท็กซี่ มอเตอร์ไซด์รับจ้าง ร้านอาหาร หรือบริการอื่น ๆ โดยจะเกิดการเรียกใช้บริการผ่านแพลตฟอร์มที่มีบริการต่าง ๆ เช่น GrabTaxi, GrabCar, GrabBike, GrabFood, GrapExpress และอื่น ๆ ทั้งนี้สรุปได้ว่าโบรกเกอร์โมเดลเป็นแบบธุรกิจที่เป็นตัวกลางที่แม่นยำในการเชื่อมโยงและส่งเสริมการซื้อขายหรือการใช้บริการระหว่างฝ่ายต่าง ๆ ผ่านแพลตฟอร์มที่เตรียมไว้ขึ้นสำหรับการทำธุรกิจนั้น

3.2 รายการโทรทัศน์และสื่อสิ่งพิมพ์ออนไลน์ (Broadcasting and Publishing)

เป็นธุรกิจที่เน้นการสร้างและนำเสนอเนื้อหาผ่านแพลตฟอร์มทางสื่อออนไลน์ เช่น Youtube ที่เป็นแพลตฟอร์มให้ผู้ผลิตรายการทีวีหรือวีดีโอบล็อก (Vlog) สร้างและแบ่งปันเนื้อหากับผู้ชม นอกจากนี้ยังรวมถึงการเผยแพร่หนังสือพิมพ์ออนไลน์ที่เป็นแพลตฟอร์มให้นักเขียนหรือคอลัมนิสต์เผยแพร่เนื้อหาผ่านอินเทอร์เน็ต เช่น Huffington Post เป็นต้น

3.3 การโฆษณา (Advertising)

การโฆษณาบนระบบออนไลน์เกิดขึ้นผ่านแพลตฟอร์มต่าง ๆ เช่น เครื่องมือสืบค้น (Search engine) เว็บพอร์ทัล (Portal web) หรือเครือข่ายสังคมออนไลน์ ซึ่งเป็นพื้นที่ที่ธุรกิจสามารถทำการโฆษณาเพื่อเชื่อมต่อกับลูกค้าได้ โดยใช้แนวคิดที่เหมาะสมกับรูปแบบและความเข้าใจของแพลตฟอร์มนั้น ๆ เช่น การ

โฆษณาผ่านการค้นหา (Search Advertising) บนเครื่องมือสืบค้น เช่น Google Ads หรือการโฆษณาบนโซเชียลมีเดีย (Social Media Advertising) ที่เป็นการโฆษณาผ่านแพลตฟอร์มสังคมออนไลน์ เช่น Facebook Ads, Instagram Ads, และ Twitter Ads

3.4 ชุมชนออนไลน์ (Online community)

เป็นแบบธุรกิจที่สร้างพื้นที่สำหรับกลุ่มคนที่มีความสนใจเดียวกันที่จะมาแลกเปลี่ยนข้อมูล แบ่งปันเนื้อหา และเชื่อมต่อกันผ่านแพลตฟอร์มออนไลน์ เช่น แพลตฟอร์มสังคมออนไลน์ที่ให้ผู้ใช้บริการสร้างโพสต์ และแบ่งปันภาพหรือวิดีโอ เช่น Flickr, YouTube และ Wiki ที่เป็นแหล่งแบ่งปันสื่อประสม นอกจากนี้ยังรวมถึงการสร้างชุมชนออนไลน์ที่เชื่อมต่อผู้คนที่มีความสนใจเหมือนกันเพื่อแบ่งปันประสบการณ์ ความรู้ และเคล็ดลับ เช่น การเป็นสมาชิก (Subscribe) และการติดตาม (Follow) บนแพลตฟอร์มทางสื่อออนไลน์ ทั้งนี้ชุมชนออนไลน์ส่งผลให้เกิดความสัมพันธ์และความเชื่อมั่นในธุรกิจและผู้ให้บริการ และส่งผลในการสร้างรายได้โดยการเป็นสมาชิกแบบจ่ายเงินหรือการขายสินค้าและบริการในชุมชนนั้น ๆ

3.5 ธุรกิจแบบอรรถประโยชน์ (Utility model)

เป็นรูปแบบของธุรกิจที่เน้นการให้บริการด้านเทคโนโลยีและสารสนเทศเพื่อสนับสนุนกิจกรรมทางธุรกิจอื่น ๆ โดยธุรกิจในแบบนี้มุ่งเน้นการให้บริการสิ่งที่มีคุณค่าเพิ่มเข้ามาเพื่อใช้ในกิจกรรมของลูกค้า รูปแบบนี้บางครั้งให้บริการในรูปแบบฟรีหรือฟรีบางส่วนเพื่อดึงดูดผู้ใช้งานและจากนั้นจะมีการเรียกเก็บค่าบริการเพิ่มเติมเมื่อผู้ใช้งานต้องการการบริการที่มีระดับคุณภาพสูงขึ้น ตัวอย่างของธุรกิจแบบอรรถประโยชน์รวมถึงการให้บริการพื้นที่เก็บข้อมูลออนไลน์ เช่น Dropbox ซึ่งให้ผู้ใช้งานใช้งานพื้นที่ฟรีในระดับหนึ่ง และเมื่อต้องการพื้นที่จัดเก็บข้อมูลเพิ่มเติมหรือความปลอดภัยเพิ่มเติมก็จะต้องจ่ายค่าสมาชิกเพิ่มเพื่อใช้บริการเสริมเติม

4. เปรียบเทียบลักษณะของธุรกิจยุคดั้งเดิมและธุรกิจยุคดิจิทัล

เกิดการเปลี่ยนแปลงที่สำคัญในวงการธุรกิจเมื่อเทคโนโลยีดิจิทัลเข้ามามีบทบาทสำคัญในการดำเนินธุรกิจ แนวคิดและลักษณะของธุรกิจก็ได้รับการเปลี่ยนแปลงตามไปด้วย สำหรับธุรกิจยุคดั้งเดิมและธุรกิจยุคดิจิทัล มีความแตกต่างกันอย่างชัดเจนดังนี้

4.1 ธุรกิจดั้งเดิม

1) มีขนาดเล็กจนถึงปานกลางและมักเริ่มจากการเป็นธุรกิจในครัวเรือนหรือบริษัทเล็ก ๆ แบบ SMEs โดยรูปแบบการจัดการมักเป็นแบบเครือญาติ

2) การทำงานในธุรกิจดั้งเดิมมักเป็นแบบมืออาชีพ คือ ผู้ที่มีความรู้และความชำนาญในการจะปฏิบัติงาน

3) เน้นการใช้แรงงานมากกว่าการใช้เทคโนโลยี การประเมินและการจัดทำงานมักพิจารณาจากสิ่งที่เกี่ยวข้องกับพื้นที่ คน และวัตถุดิบ

4) ใช้เทคโนโลยีการสื่อสารพื้นฐาน เช่น จดหมาย โทรสาร และโทรศัพท์ เป็นรูปแบบการติดต่อสื่อสารหลัก

5) การบันทึกข้อมูลมักใช้วิธีการจดจำและบันทึกลงสมุด

6) การบริหารงานในธุรกิจดั้งเดิมจะใช้ระบบเส้นสายในการสื่อสาร และการตัดสินใจมักพิจารณาจากความคิดของตัวเองมากกว่าข้อมูล

7) ธุรกิจดั้งเดิมมักขาดยุทธศาสตร์เชิงรุกที่เน้นไปที่การนำเสนอสิ่งที่เป็นนวัตกรรมหรือการเปลี่ยนแปลงเชิงโครงสร้าง

4.2 ธุรกิจยุคดิจิทัล

1) ธุรกิจยุคดิจิทัลใช้เทคโนโลยีขั้นสูงเข้ามาช่วยในการดำเนินงานและสร้างความเชื่อมั่นในผู้ใช้งาน

2) การสื่อสารและการทำงานมักเกิดขึ้นทันทีผ่านเครือข่ายออนไลน์ ซึ่งทำให้การติดต่อและการทำงานเป็นไปได้รวดเร็วและมีประสิทธิภาพ

3) การจัดเก็บข้อมูลมักใช้ฐานข้อมูลคอมพิวเตอร์เพื่อเก็บข้อมูลและการประมวลผล

4) การบริหารงานในธุรกิจยุคดิจิทัลมักใช้ข้อมูลและข่าวสารในการตัดสินใจ และมีการกำหนดยุทธศาสตร์เชิงรุกเพื่อนำเสนอนวัตกรรมและทำการปรับเปลี่ยนตามสถานการณ์

5) ธุรกิจยุคดิจิทัลมักเน้นการนำเอาเทคโนโลยีมาใช้เพื่อสร้างความมั่นคงและความปลอดภัยในการดำเนินงานและการทำธุรกิจ

ทั้งสองรูปแบบของธุรกิจมีลักษณะและวิถีชีวิตที่แตกต่างกันอย่างชัดเจน การเปลี่ยนแปลงของเทคโนโลยีและวัฒนธรรมออนไลน์ได้นำไปสู่การเคลื่อนไหวของวงการธุรกิจและการทำงานในทั้งสองรูปแบบอย่างต่อเนื่อง

สรุปท้ายบทที่ 1

ข้อมูล (Data) หมายถึง ข้อเท็จจริงเกี่ยวกับเหตุการณ์ที่เกิดขึ้นซึ่งเป็นข้อมูลดิบ (Raw Data) ที่ยังการประมวลผล จึงยังไม่มีความหมายในการนำไปใช้งาน

สารสนเทศ (Information) หมายถึง ข้อมูลที่ผ่านการประมวลผลเพื่อนำไปใช้ประโยชนในด้านการจัดการอย่างใดอย่างหนึ่ง สารสนเทศที่ดีจะต้องมีความถูกต้องและเที่ยงตรง มีความสมบูรณ์ ทันต่อเวลาที่ต้องการใช้ ตรงกับความต้องการใช้งานของผู้ใช้ มีความทันสมัย และสามารถตรวจสอบแหล่งที่มาของข้อมูลได้

ระบบสารสนเทศ (Information System) หมายถึง กลุ่มขององค์ประกอบทางด้านสารสนเทศที่มีความสัมพันธ์กันและสามารถทำงานร่วมกันเพื่อตอบสนองความต้องการใช้งานในแต่ละองค์กร ภายในระบบสารสนเทศหนึ่ง ๆ ยังสามารถแบ่งได้เป็นระบบย่อย (Subsystem) ที่มีความสัมพันธ์กัน ระบบสารสนเทศของแต่ละองค์จึงมีรายละเอียดที่แตกต่างกันได้ทั้งนี้ขึ้นอยู่กับขอบเขต (Boundary) และสิ่งแวดล้อม (Environment) ของระบบสารสนเทศนั้น ๆ ระบบสารสนเทศสามารถแบ่งได้เป็น 6 ประเภทหลัก ตามวัตถุประสงค์และคุณลักษณะของระบบ ดังนี้

1. ระบบประมวลสารสนเทศเชิงรายการ
2. ระบบสารสนเทศเพื่อการจัดการ
3. ระบบสารสนเทศเพื่อสนับสนุนการตัดสินใจ
4. ระบบสารสนเทศสำหรับผู้บริหาร
5. ระบบสำนักงานอัตโนมัติ
6. ระบบจัดการความรู้

นวัตกรรม (Innovation) หมายถึง กระบวนการหรือผลิตภัณฑ์ใหม่ที่สร้างขึ้นโดยมีการนำเสนอแนวคิดใหม่ หรือการปรับเปลี่ยนแนวคิดที่มีอยู่ให้มีคุณค่าและสร้างสรรค์ หรือการสร้างผลิตภัณฑ์หรือบริการใหม่เพื่อตอบสนองความต้องการของตลาดหรือกลุ่มลูกค้าใหม่ ซึ่งเมื่อมีคุณค่าและมีประโยชน์แล้วจะสามารถขยายผลต่อได้เชิงพาณิชย์ หรือขายได้นั่นเอง สามารถแบ่งเป็น 4 ประเภท ดังนี้

1. นวัตกรรมแบบก่อกวนทำลาย (Disruptive Innovation)
2. นวัตกรรมแบบสิ้นเชิง (Radical Innovation)
3. นวัตกรรมแบบยั่งยืน (Sustaining Innovation)
4. นวัตกรรมแบบต่อยอด (Incremental Innovation)

เทคโนโลยี คือ วิทยาการที่เกี่ยวกับการนำเอาวิทยาศาสตร์มาประยุกต์ใช้ให้เกิดประโยชน์ในทางปฏิบัติและอุตสาหกรรม หรือเป็นการประยุกต์ความรู้ด้านวิทยาศาสตร์ให้ใช้เพื่อแก้ไขปัญหาต่าง ๆ และสร้างสรรค์วัสดุอุปกรณ์ เครื่องมือ และเครื่องจักร รวมถึงการสร้างความรู้ใหม่ ๆ เพื่อให้การดำรงชีวิตของมนุษย์เป็นไปอย่างง่ายและสะดวกยิ่งขึ้น

เทคโนโลยีมีลักษณะที่สามารถจำแนกได้เป็น 3 ลักษณะ

1. เทคโนโลยีในลักษณะของกระบวนการ (Process Technology)
2. เทคโนโลยีในลักษณะของผลผลิต (Product Technology)
3. เทคโนโลยีในลักษณะผสมของกระบวนการและผลผลิต (Process and Product Technology)

รูปแบบธุรกิจดิจิทัลมุ่งเน้นการแบ่งปันมูลค่าในรูปแบบเศรษฐกิจแบ่งปัน (Shared Economy) และการดำเนินกิจกรรมในลักษณะปัจจุบันหรือเรียลไทม์บนแพลตฟอร์ม (Live or Real-time Market) โดยธุรกิจดิจิทัลจะใช้แพลตฟอร์ม (Platform) รูปแบบธุรกิจแบบแพลตฟอร์ม มักมีลักษณะต่อไปนี้

1. โบรกเกอร์โมเดล (Brokerage model)
2. รายการโทรทัศน์และสื่อสิ่งพิมพ์ออนไลน์ (Broadcasting and Publishing)
3. การโฆษณา (Advertising)
4. ชุมชนออนไลน์ (Online community)
5. ธุรกิจแบบอรรถประโยชน์ (Utility model)

คำถามท้ายบทที่ 1

1. อธิบายความหมายและองค์ประกอบของระบบสารสนเทศ
2. ระบบสารสนเทศสามารถแบ่งตามวัตถุประสงค์และคุณลักษณะสำคัญกี่ประเภท จงอธิบายแต่ละประเภทมาพอสังเขป
3. อธิบายความหมายของนวัตกรรม และอธิบายประเภทของนวัตกรรม อธิบายโดยสังเขป
4. อธิบายความหมายของเทคโนโลยี และอธิบายลักษณะของเทคโนโลยี อธิบายโดยสังเขป
5. ธุรกิจสมัยใหม่คืออะไร จงอธิบาย
6. ลักษณะของธุรกิจดิจิทัลคืออะไร อธิบายโดยสังเขป
7. ยกตัวอย่างการเปรียบเทียบลักษณะของธุรกิจยุคดั้งเดิมและธุรกิจยุคดิจิทัล มา 3 ตัวอย่าง
8. รูปแบบธุรกิจแบบแพลตฟอร์มคืออะไร อธิบายโดยสังเขป

บทที่ 2
วงจรของการพัฒนาระบบ

การพัฒนาระบบสารสนเทศ หมายถึง การสร้างระบบสารสนเทศใหม่ หรือการปรับปรุงระบบสารสนเทศเดิมให้มีประสิทธิภาพมากยิ่งขึ้น โดยการประยุกต์ใช้ระบบคอมพิวเตอร์เป็นเครื่องมือหลักในการทำงาน การพัฒนาระบบในองค์กรมักจัดทำในรูปแบบของโครงการพัฒนาระบบภายใต้แผนการเนินงาน และแผนการใช้ทรัพยากรที่ได้กำหนดไว้โดยมีทรัพยากรหลัก คือ บุคลากรที่เกี่ยวข้องกับการพัฒนาระบบนั้น ๆ จะต้องเสียสละเวลาเข้าร่วมในขั้นตอนต่าง ๆ ของการพัฒนาระบบเพื่อให้ได้ระบบสารสนเทศที่สอดคล้องต่อความต้องการใช้งานขององค์กรและมีประสิทธิภาพตรงตามเป้าหมายที่กำหนดไว้

วงจรการพัฒนาระบบ (System Development Life Cycle: SDLC)

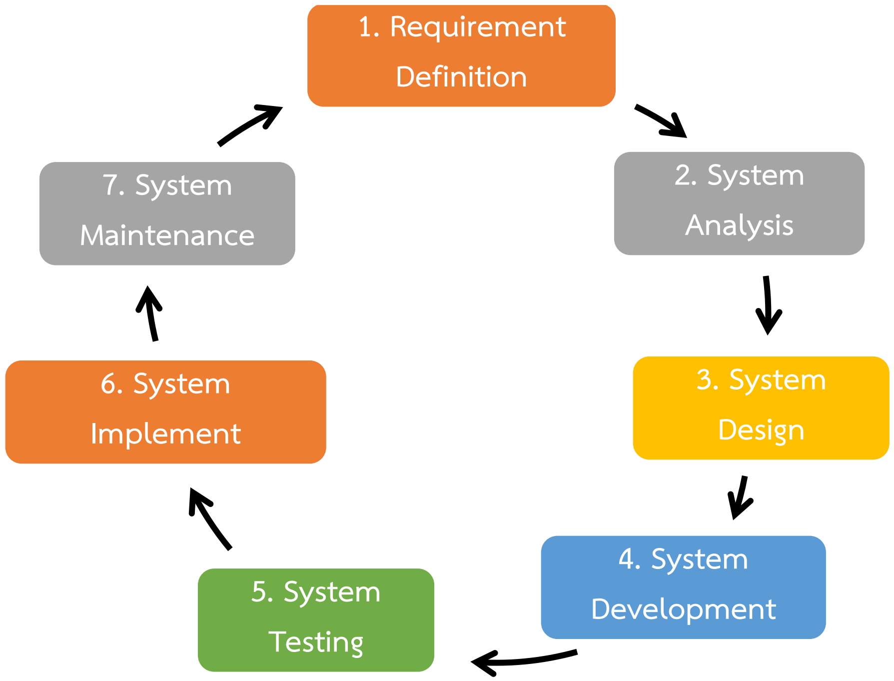
รูปแบบของวงจรการพัฒนาระบบ (Software Development Life Cycle: SDLC) มีหลายรูปแบบที่สามารถใช้ในการพัฒนาซอฟต์แวร์ ซึ่งแต่ละรูปแบบมีลักษณะและขั้นตอนที่แตกต่างกันเพื่อตอบสนองความต้องการและความซับซ้อนของโครงการ

การพัฒนาระบบสารสนเทศเป็นสิ่งจำเป็นต้องอาศัยความรู้ ความตั้งใจ และความร่วมมือจากทุกฝ่ายในหน่วยงาน จึงจะประสบผลสำเร็จ กระบวนการหรือขั้นตอนในการพัฒนาระบบสารสนเทศจึงมีความสัมพันธ์กับการดำเนินงานในองค์กรเป็นอย่างมาก และมีปฏิสัมพันธ์ต่อกันด้วย เช่น การพัฒนาระบบสารสนเทศในระดับปฏิบัติการ ซึ่งโดยปกติมีการดำเนินงานที่มีรูปแบบชัดเจนและปฏิบัติเป็นประจำอยู่แล้ว ขั้นตอนในการสร้างระบบสารสนเทศจะเริ่มด้วยการวิเคราะห์ระบบงานที่เป็นอยู่ประจำแล้วนำมาออกแบบระบบสารสนเทศที่สอดคล้องกับลักษณะงาน (Theeraviriya et al., 2019) ทดสอบระบบว่าสามารถทำงานได้ตามที่ต้องการหรือไม่ จากนั้นจะปรับเปลี่ยนจากระบบ เดิมเข้าสู่ระบบใหม่ที่ปฏิบัติงานด้วยคอมพิวเตอร์ ทำการติดตามผลที่เกิดขึ้นจากการใช้ระบบปรับระบบ สารสนเทศให้เหมาะสมยิ่งขึ้น

การพัฒนาระบบสารสนเทศ เป็นกระบวนการในการสร้างระบบสารสนเทศขึ้นมาเพื่อใช้แก้ปัญหา หรือสร้างมูลค่าเพิ่มให้กับหน่วยงาน การพัฒนาระบบสารสนเทศรวมถึงการปรับเปลี่ยนระบบงานเดิมที่มีอยู่แล้ว ให้สามารถทำงานแก้ปัญหาการดำเนินงานตามความต้องการของผู้ใช้งาน โดยอาจนำเอาระบบคอมพิวเตอร์เข้ามาช่วยประมวลผล เรียบเรียง เปลี่ยนแปลงและจัดเก็บข้อมูล เพื่อให้ได้สารสนเทศที่ถูกต้อง และเป็นการเพิ่มประสิทธิภาพในการทำงานขององค์กรอีกด้วย ทั้งนี้งานพัฒนาระบบสารสนเทศจะมีกิจกรรมและขั้นตอนต่าง ๆ มากมาย รวมถึงความซับซ้อนของระบบงาน ดังนั้น การมีแนวทางที่เป็นลำดับขั้นตอนที่ส่งผลต่อมาตรฐานของ

ระบบงานจึงเป็นสิ่งที่ในการวิเคราะห์ระบบต้องการก็เพื่อให้งานพัฒนาระบบเป็นไปในทิศทางเดียวกันมีขั้นตอนลำดับกิจกรรมที่ต้องทำอย่างชัดเจนในแต่ละขั้นตอน จึงเกิด "วงจรการพัฒนาระบบ" ขึ้นมา

Kaewman, & Akararungruangkul (2018) วงจรการพัฒนาระบบ เป็นกระบวนการทางความคิดในการพัฒนาระบบสารสนเทศเพื่อแก้ปัญหาและตอบสนองความต้องการของผู้ใช้ หรือองค์กรนั้น ๆ ภายในวงจรการพัฒนานั้นจะแบ่งกระบวนการพัฒนาเป็นขั้นตอน โดยทั่วไปวงจรการพัฒนาระบบจะเริ่มต้นที่ 5 - 7 ขั้นตอน และในวงจรการพัฒนา 3 - 4 ขั้นตอนแรกจะเป็นการวิเคราะห์และออกแบบระบบ ซึ่งเป็นส่วนที่สำคัญมากในการพัฒนาระบบขึ้นอยู่กับขนาดของโครงการ และความซับซ้อนของโครงการด้วย ในปัจจุบันการพัฒนาระบบมีความซับซ้อนเพิ่มมากขึ้น ในหัวข้อนี้จะอธิบายวงจรการพัฒนาระบบ 7 ขั้นตอน ดังภาพที่ 2.1



ภาพที่ 2.1 วงจรการพัฒนาระบบ SDLC

1. ขั้นตอนที่หนึ่ง การกำหนดความต้องการของระบบ (Requirement Definition Phase)

ในขั้นตอนนี้ผู้พัฒนาระบบจะต้องค้นหาปัญหาและศึกษาทำความเข้าใจปัญหาข้อเท็จจริงต่าง ๆ ที่เกิดขึ้นจากการทำงานในระบบงานเดิม โดยผู้พัฒนาระบบจะต้องทำความเข้าใจปัญหาอย่างถ่องแท้ คิดหาแนวทางและวัตถุประสงค์ในการแก้ปัญหา ศึกษาความเป็นไปได้ในการแก้ปัญหารวบรวมความต้องการและสรุปข้อกำหนดให้ชัดเจน ถูกต้องและเป็นที่ยอมรับทั้งสองฝ่ายพร้อมทั้งกำหนดแผนการดำเนินกิจกรรม ซึ่งแบ่งออกเป็น 3 ขั้นตอนย่อย คือ

1.1) การกำหนดปัญหา (Problem Definition) เป็นขั้นตอนที่ผู้พัฒนาระบบเข้าไปทำความเข้าใจปัญหา คือการทำความเข้าใจกับปัญหาที่เกิดขึ้นอย่างถ่องแท้ว่าปัญหาที่แท้จริงคืออะไรพร้อมทั้งคิดหาแนวทางหรือสถานการณ์ที่ผู้พัฒนาระบบเชื่อว่าสามารถปรับปรุงระบบให้ดีขึ้นและผู้พัฒนาระบบต้องค้นหาว่า การดำเนินงานนั้นต้องการอะไร เพื่อทำให้เป้าหมายบรรลุวัตถุประสงค์ ในขั้นตอนนี้ผู้พัฒนาระบบจะทำการเก็บรวบรวมข้อมูลจากผู้ใช้ระบบ โดยการตรวจสอบเอกสาร การสัมภาษณ์ การออกแบบสอบถาม หรือสังเกต พฤติกรรมและสภาพแวดล้อมของการดำเนินงาน

1.2) การศึกษาความเป็นไปได้ (Feasibility Study) เป็นขั้นตอนที่ผู้พัฒนาระบบทำการศึกษาความเป็นไปได้ในการดำเนินการปรับปรุงระบบ พิจารณาถึงความพร้อมในด้านต่าง ๆ รวมถึงความเสี่ยงที่อาจส่งผลกระทบต่อความล้มเหลวในการปรับปรุงระบบ โดยทำการศึกษาความเป็นไปได้ ในแง่มุมต่าง ๆ ดังต่อไปนี้

1.2.1) ความเป็นไปได้ทางเทคนิค คือความเป็นไปได้ของการปรับปรุงระบบหรือพัฒนาระบบใหม่ โดยนำเทคโนโลยีปัจจุบันมาใช้งานหรือการยกระดับเครื่องคอมพิวเตอร์หรือตัดสินใจใช้เทคโนโลยีใหม่ทั้งหมด

1.2.2) ความเป็นไปได้ทางเศรษฐศาสตร์ คือความเป็นไปได้ในเชิงเศรษฐศาสตร์ด้วยการคำนึงถึงต้นทุนค่าใช้จ่ายในการพัฒนาระบบงาน ความคุ้มค่าของระบบด้วยการเปรียบเทียบผลลัพธ์ที่ได้ จากระบบกับค่าใช้จ่ายที่ต้องลงทุน

1.2.3) ความเป็นไปได้ด้านการปฏิบัติงาน คือความเป็นไปได้ของระบบใหม่ที่จะให้สารสนเทศที่ถูกต้องตรงความต้องการของผู้ใช้งานทัศนคติ ทักษะกับระบบงานใหม่ที่มีการปรับเปลี่ยน โครงสร้างการทำงานใหม่ว่าเป็นที่ยอมรับหรือไม่

1.2.4) ความเป็นไปได้ทางด้านเวลาในการดำเนินการ คือความเป็นไปได้ของระยะเวลา ในการดำเนินงานในการพัฒนาระบบใหม่

1.3) การกำหนดความต้องการของระบบ (System Requirements) เป็นขั้นตอนที่ผู้พัฒนาระบบทำการวิเคราะห์การทำงานระบบเดิม เพื่อค้นหาปัญหาที่เกิดขึ้นและรวบรวมรายละเอียดต่าง ๆ เพื่อกำหนดจุดประสงค์ในการหาข้อสรุปในด้านของความต้องการระหว่างผู้พัฒนากับผู้ใช้งาน เรียกว่าข้อกำหนดความต้องการ (Requirement Specification) เพื่อใช้ในการพัฒนาระบบโดยผู้พัฒนาระบบจะต้องนำข้อกำหนดความต้องการเสนอต่อผู้บริหาร เพื่อพิจารณาและตัดสินใจในการดำเนินการพัฒนาระบบหรือล้มเลิกการพัฒนาระบบ ดังที่ (นาฏวราพร สิมทอง, 2552) กล่าวว่า เป็นการศึกษาถึงขั้นตอนการดำเนินงานของระบบเดิม หรือระบบปัจจุบันว่าเป็นอย่างไรบ้างปัญหาที่เกิดขึ้นคืออะไร หลังจากนั้นจึงรวบรวมความต้องการในระบบใหม่

2. ขั้นตอนที่สอง การวิเคราะห์ระบบ (System Analysis Phase)

เป็นขั้นตอนการวิเคราะห์ระบบงานปัจจุบันในลักษณะของการพัฒนาแนวคิดสำหรับระบบใหม่ วัตถุประสงค์หลักในการวิเคราะห์ระบบ คือต้องศึกษาและทำความเข้าใจเกี่ยวกับความต้องการต่าง ๆ ตามที่ได้รวบรวมจากขั้นตอนการกำหนดความต้องการของระบบแล้วทำการวิเคราะห์เพื่อประเมินว่าระบบใหม่ควรมีหน้าที่อะไรบ้างที่ต้องดำเนินการ หรืออาจกล่าวได้ว่าเป็นการนำความต้องการของผู้ใช้ที่ได้มาจากขั้นตอนแรกมาวิเคราะห์ในรายละเอียดเพื่อทำการพัฒนาเป็นแบบจำลองเชิงตรรกะ (Logical Model) โดยขั้นตอนการวิเคราะห์ระบบอย่างละเอียดจะกล่าวถึงในบทที่ 3 โดยขั้นตอนนี้มีอยู่ 2 ขั้นตอนหลัก ดังนี้

2.1) การรวบรวมความต้องการใช้งานของผู้ใช้งาน (User Requirement) ตั้งแต่ผู้บริหาร หัวหน้าแผนกและเจ้าหน้าที่ทุกฝ่ายที่เกี่ยวข้อง ซึ่งนักวิเคราะห์ระบบจะต้องรวบรวมความต้องการในส่วนนี้ให้ครบถ้วนเพื่อนำมาสรุปเป็นความต้องการของระบบ โดยอาศัยวิธีการรวบรวมข้อมูลหลายวิธีการประกอบการเพื่อให้ได้ข้อมูลที่ครบถ้วนทุกประเด็นตามขอบเขตโครงการ

1) ศึกษาจากเอกสารที่เกี่ยวข้องกับการปฏิบัติงาน

2) การค้นคว้าข้อมูลจากแหล่งต่าง ๆ

3) การสัมภาษณ์

4) การใช้แบบสอบถาม

5) ค่าสังเกตการณ์ในระหว่างการปฏิบัติงานของผู้เกี่ยวข้องกับระบบ

2.2) การวิเคราะห์ข้อมูลความต้องการของระบบ (System Requirement) จากผลสรุปความต้องการใช้งานระบบที่ได้รวบรวมไว้แล้ว นักวิเคราะห์ระบบพิจารณาว่าความต้องการส่วนใดที่ควรพัฒนาเป็นระบบสารสนเทศด้วยความต้องการส่วนใดที่ไม่เหมาะสม หรือไม่มีความสำคัญเร่งด่วนในการพัฒนาเป็นฟังก์ชันหรือการใช้งานในโครงการนี้ โดยจะต้องสอดคล้องกับสัญญาว่าจ้างของโครงการและรายงานการศึกษาเบื้องต้นที่ได้จัดทำไว้ก่อนหน้านี้ ซึ่งเอกสารสำคัญที่ได้จากขั้นตอนนี้คือ ข้อกำหนดเกี่ยวกับความต้องการของระบบ (System Requirement Specification: SRS) เพื่อสร้างความเข้าใจร่วมกันระหว่างทีมงานพัฒนาระบบ รวมไปถึงผู้ที่มีส่วนเกี่ยวข้องกับระบบนักวิเคราะห์ระบบมักจะมีการสร้างแบบจำลองเป็นภาพที่แสดงให้เห็นถึงความสำคัญขององค์ประกอบต่าง ๆ ของระบบ ประกอบด้วยแบบจำลองกระบวนการ (Process Model) และแบบจำลองข้อมูล (Data Model) เช่น แผนภาพกระแสข้อมูล (Data Flow Diagram: DFD) ที่จะกล่าวถึงในบทที่ 4 และแผนภาพ E-R ที่จะกล่าวถึงในบทที่ 5 ต่อไป

3. ขั้นตอนที่สาม การออกแบบระบบ (System Design Phase)

ในขั้นตอนนี้ ผู้พัฒนาระบบจะออกแบบระบบสารสนเทศที่สอดคล้องกับความต้องการที่ได้ระบุไว้ในเอกสารขั้นตอนของการวิเคราะห์ระบบซึ่งเป็นแบบจำลองเชิงตรรกะ (Logical Model) มาพัฒนาเป็นการออกแบบเชิงเชิงกายภาพ (Physical Design) โดยแบบจำลองเชิงตรรกะจะมุ่งเน้นว่ามีอะไรบ้างที่ต้องทำในระบบในขณะที่การออกแบบเชิงเชิงกายภาพจะนำแบบจำลองเชิงตรรกะมาพัฒนาต่อด้วยการมุ่งเน้นว่าระบบจะดำเนินงานอย่างไรเพื่อให้เกิดผลตามความต้องการ โดยระบุถึงคุณลักษณะของอุปกรณ์และเทคโนโลยีที่จะนำมาใช้โปรแกรมภาษาที่จะนำมาพัฒนาการออกแบบรายงาน การออกแบบหน้าจอในการติดต่อกับผู้ใช้ระบบ การออกแบบรูปแบบข้อมูลที่นำเข้าและรูปแบบการรับข้อมูลการออกแบบผังระบบงาน การออกแบบฐานข้อมูล สิ่งที่ได้จากการออกแบบเชิงกายภาพนี้จะเป็นข้อมูลเฉพาะของการออกแบบ (System Design Specification)

ขั้นตอนการออกแบบเชิงกายภาพเป็นการนำผลจากการออกแบบระบบเชิงตรรกะมาระบุลักษณะการทำงานของระบบทางกายภาพโดยประกอบด้วย

3.1) การกำหนดคนลักษณะเฉพาะของฮาร์ดแวร์และซอฟต์แวร์ที่เหมาะสม ในส่วนของความต้องการขั้นต่ำของฮาร์ดแวร์ ความต้องการด้านซอฟต์แวร์ และการบำรุงรักษาหลังการติดตั้งใช้งาน

3.2) การออกแบบฐานข้อมูลเข้าระบบ เป็นการกำหนดรายละเอียดและโครงสร้างฐานข้อมูลสำหรับรองรับการทำงานของระบบที่พัฒนา

3.3) การออกแบบคุณลักษณะเฉพาะของโปรแกรม โดยทั่วไปมักอยู่ในรูปแบบของซูโดโค๊ด (Pseudocode) ผังงานโปรแกรม (Program Flowchart) และผังงานระบบ (System Flowchart) ซึ่งในขั้นตอนนี้นักวิเคราะห์ระบบสามารถจัดเตรียมข้อมูลเบื้องต้นสำหรับทดสอบระบบตามที่ได้ออกแบบคุณลักษณะเฉพาะของโปรแกรมไว้ รวมไปถึงการวางแผนในการจัดทำสื่อประกอบการฝึกอบรมการใช้งานระบบหลังจากที่ได้พัฒนาระบบเสร็จสมบูรณ์แล้ว

3.4) การออกแบบระบบรักษาความปลอดภัย ทั้งในส่วนของสถานที่ติดตั้งระบบคอมพิวเตอร์ ส่วนของเครื่องคอมพิวเตอร์ ส่วนของข้อมูล และส่วนของระบบเครือข่าย

ในปัจจุบันนักวิเคราะห์ระบบมักจะสร้างโปรแกรมต้นแบบ (Prototype) เป็นแบบจำลองของระบบที่จะพัฒนาเพื่อนำเสนอให้ผู้ใช้ได้มองเห็นภาพรวมการทำงานทั้งระบบ โดยผู้ใช้สามารถแสดงความคิดเห็นร่วมกันว่าโปรแกรมต้นแบบที่สร้างขึ้น มีส่วนใดที่ต้องการให้มีการปรับรูปแบบหรือวิธีการใช้งานหรือไม่ ก่อนดำเนินการพัฒนาระบบ ในเนื้อหาการออกแบบส่วนติดต่อผู้ใช้จะกล่าวถึงอย่างละเอียดในบทที่ 6 ต่อไป

4. ขั้นตอนที่สี่ การพัฒนาระบบ (System Development Phase)

เป็นขั้นตอนที่นำเอาระบบที่ได้ออกแบบไว้มาทบทวนเพื่อพัฒนาโปรแกรม และการทดสอบโปรแกรม ในกระบวนการนี้นักพัฒนาระบบจะต้องพัฒนาโปรแกรมให้เป็นไปตามคุณลักษณะและรูปแบบตามที่นักวิเคราะห์ระบบได้กำหนดไว้ การเขียนชุดคำสั่งเพื่อสร้างระบบงานทางคอมพิวเตอร์โดยนักพัฒนาระบบสามารถนำเครื่องมือเข้ามาช่วยในการพัฒนาโปรแกรม เพื่อช่วยให้ระบบงานสามารถพัฒนาได้เร็วขึ้นและมีคุณภาพ และในกระบวนการนี้จะต้องจัดทำเอกสารโปรแกรมควบคู่ไปกับการพัฒนาโปรแกรมเพื่อทำให้ง่ายต่อการตรวจสอบและแก้ไขข้อกำหนดเกี่ยวกับการพัฒนาโปรแกรม ผู้บริหารขององค์กรจะเป็นผู้ตัดสินใจเลือกวิธีการพัฒนาโปรแกรม บางองค์กรอาจมีทีมพัฒนาโปรแกรมในองค์กร หรือจัดซื้อซอฟต์แวร์สำเร็จรูปมาใช้ หรือจ้างบริษัทที่รับพัฒนาระบบโดยเฉพาะ

5. ขั้นตอนที่ห้า การทดสอบระบบ (System Testing Phase)

เมื่อได้พัฒนาโปรแกรมแล้ว ยังไม่สามารถนำระบบไปใช้งานได้ทันทีจำเป็นต้องดำเนินการทดสอบระบบก่อนที่จะนำระบบไปใช้งานจริง การทดสอบเบื้องต้นด้วยการสร้างข้อมูลจำลองขึ้นมาเพื่อตรวจสอบการทำงานของระบบงาน หากพบข้อผิดพลาดจึงทำการปรับปรุงแก้ไขให้ถูกต้อง โดยการทดสอบระบบนี้จะมีการตรวจสอบอยู่ 2 ส่วน คือ การตรวจสอบรูปแบบภาษาเขียน (Syntax) และการตรวจสอบการทำงานของระบบว่าตรงกับความต้องการของผู้ใช้หรือไม่

6. ขั้นตอนที่หก การติดตั้งระบบ (System Implement Phase)

เมื่อทำการทดสอบระบบจนมั่นใจว่าระบบสามารถทำงานได้จริง และตรงกับความต้องการของผู้ใช้ระบบตลอดจนพร้อมที่จะนำไปติดตั้งเพื่อใช้งานในสถานการณ์จริง การติดตั้งระบบคือการเปลี่ยนการทำงานจากระบบงานเดิมหรือสภาพแวดล้อมของการพัฒนาระบบ (System Environment) ไปเป็นระบบงานใหม่ที่มีสภาพแวดล้อมของระบบพร้อมทำงานจริง แต่การเปลี่ยนแปลงไปสู่สภาพแวดล้อมใหม่ย่อมมีผลกระทบต่อผู้ใช้งานบางกลุ่มที่ยังคงมีความคุ้นเคยกับวิธีการดำเนินงานแบบเก่า รวมทั้งข้อจำกัดในเรื่องของความพร้อมในการเปลี่ยนแปลง ดังนั้นจึงควรเลือกแนวทางที่เหมาะสมในการติดตั้งด้วย ซึ่งแบ่งออกเป็น 4 แนวทาง ดังต่อไปนี้

6.1) การติดตั้งแบบทันทีทันใด (Direct Installation) เป็นวิธีการติดตั้งระบบใหม่ทันทีและยกเลิกการใช้งานระบบเก่าทันทีเช่นเดียวกัน

6.2) การติดตั้งแบบขนาน (Parallel Installation) เป็นวิธีการติดตั้งระบบใหม่ไปพร้อม ๆ กับการใช้งานระบบเก่า จนกว่าผู้ใช้และผู้บริหารจะมีความพอใจระบบใหม่จึงตัดสินใจหยุดใช้งานระบบเก่า

6.3) การติดตั้งแบบนำร่อง (Single Location Installation/Pilot Installation) เป็นวิธีการติดตั้งที่มีการใช้งานระบบงานใหม่เพียงหน่วยเดียวขององค์กรก่อนเพื่อเป็นการนำร่องแล้วจึงค่อยปรับเปลี่ยน ทั้งหมดเมื่อเห็นว่าระบบใหม่นั้นลงตัวแล้ว

6.4) การติดตั้งแบบทยอยติดตั้งเป็นระยะ (Phased Installation) เป็นวิธีการที่ติดตั้งระบบใหม่เพียงบางส่วนก่อนระยะหนึ่งควบคู่ไปกับการใช้งานระบบเก่า แล้วจึงค่อย ๆ ทยอยใช้ระบบงานใหม่เพิ่มขึ้น ทีละส่วนจนกระทั่งครบทุกส่วนของระบบงานใหม่อย่างเต็มรูปแบบในการใช้งานระบบใหม่ทดแทนระบบงานเดิม ผู้พัฒนาระบบจำเป็นต้องมีการแปลงข้อมูลจากระบบงานเดิมให้มาอยู่ในรูปแบบที่ระบบใหม่สามารถนำไปใช้งานได้ และในขั้นตอนนี้ผู้พัฒนาระบบจะต้องดำเนินการจัดทำเอกสารคู่มือระบบรวมถึงการฝึกอบรมผู้ใช้ระบบ

7. ขั้นตอนที่เจ็ด การบำรุงรักษาระบบ (System Maintenance Phase)

หลังจากที่ระบบงานที่พัฒนาขึ้นมาใหม่ได้ถูกนำไปใช้งานเป็นที่เรียบร้อยแล้ว หากพบข้อผิดพลาดหรือข้อบกพร่องจากการทำงานของระบบงานใหม่ ผู้พัฒนาระบบจึงจำเป็นต้องดำเนินการติดตามและแก้ไขให้ถูกต้อง รวมถึงกรณีข้อมูลที่จัดเก็บมีปริมาณมากขึ้น การขยายระบบเครือข่ายเพื่อรองรับเครื่องลูกข่ายที่มีจำนวนมากขึ้น บางกรณีอาจจำเป็นต้องเขียนโปรแกรมเพิ่มเติมหากผู้ใช้มีความต้องการเพิ่มขึ้น ในขั้นตอนนี้จะประกอบไปด้วยกิจกรรมหลัก ๆ ที่จะต้องบำรุงรักษา ได้แก่ การบำรุงรักษาระบบ (System Maintenance) การเพิ่มเติมคุณสมบัติใหม่ ๆ เข้าไปในระบบ (Enhance the System) และการสนับสนุนงานของผู้ใช้ (Support the Users) โดยทั่วไปสัญญาว่าจ้างพัฒนาระบบมักกำหนดระยะเวลาการบำรุงรักษาระบบโดยไม่เสียค่าใช้จ่ายไว้ 1 ปี หลังจากนั้นถ้าองค์กรต้องการให้บำรุงรักษาเพิ่มเติมอาจมีค่าใช้จ่ายเพิ่มขึ้นแล้วแต่กรณี

แบบจำลองของวงจรการพัฒนาระบบ (System Development Life Cycle Model : SDLC Model)

แบบจำลองของวงจรการพัฒนาระบบ เป็นแผนภาพที่แสดงถึงรูปแบบการดำเนินงานของการพัฒนาระบบที่มีผู้คิดค้นไว้หลายรูปแบบ ซึ่งแต่ละรูปแบบจะมีจุดเด่นและความเหมาะสมในการนำไปประยุกต์ใช้งานกับการพัฒนาระบบที่แตกต่างกัน ในหัวข้อนี้ขอยกตัวอย่างแบบจำลองของวงจรการพัฒนาระบบ SDLC 5 แบบจำลองดังต่อไปนี้

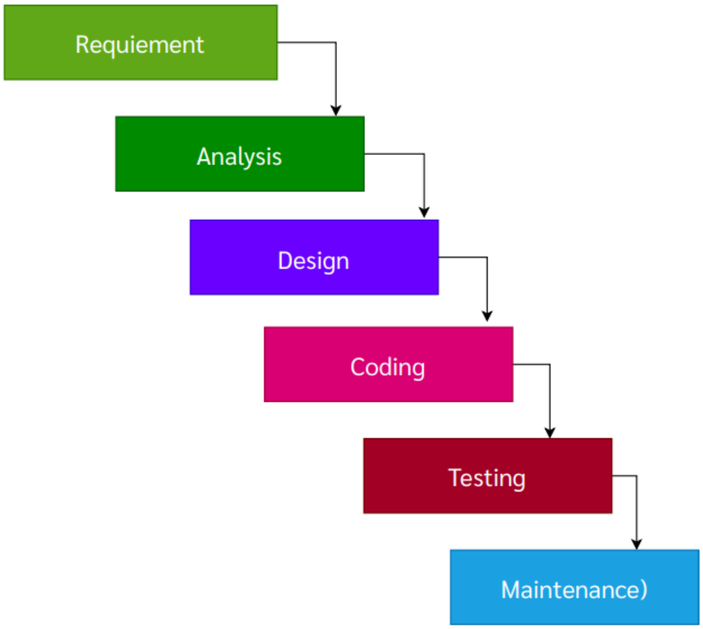
1. แบบจำลองน้ำตก (Waterfall Model)

แบบจำลองน้ำตก (Waterfall Model) เป็นแบบจำลองในการพัฒนาซอฟต์แวร์ที่เคยได้รับความนิยมในอดีตตั้งแต่ปี ค.ศ. 1970 ซึ่งมีลักษณะเป็นกระบวนการที่เรียงต่อกันแบ่งเป็นขั้นตอนตามลำดับ แต่มีลักษณะที่ไม่

สามารถย้อนกลับไปแก้ไขข้อผิดพลาดที่เกิดขึ้นในขั้นตอนที่ผ่านมาแล้วได้ เนื่องจากการทำงานในแต่ละขั้นตอนจะถูกผลิตภัณฑ์ของขั้นตอนก่อนหน้าเป็นข้อมูลเข้า เมื่อเข้าสู่ขั้นตอนใหม่แล้วก็ไม่สามารถกลับไปแก้ไขได้ ดังนั้นกระบวนการนี้มักจะใช้สำหรับโครงการที่มีความชัดเจนและแน่นอนในข้อมูลและความต้องการเริ่มต้น (อรยา ปรีชาพานิช, 2557)

ขั้นตอนในแบบจำลองน้ำตกมักจะเริ่มต้นด้วยการวางแผน การวางแผนจะทำให้เกิดความเข้าใจในขอบเขตของโครงการ และกำหนดค่ายุติธรรมของผลงาน ต่อมาคือขั้นตอนการออกแบบที่จะทำการออกแบบระบบตามความต้องการที่ได้กำหนดไว้ จากนั้นคือขั้นตอนการสร้างและพัฒนาระบบ และในขั้นตอนนี้จะมีการทดสอบระบบเพื่อตรวจสอบว่าระบบทำงานตามความต้องการหรือไม่ ขั้นตอนสุดท้ายคือการนำระบบที่พัฒนาเสร็จสิ้นไปใช้งานจริง และมีการดูแลรักษาและบำรุงรักษาระบบในระยะยาว

อย่างไรก็ตาม รูปแบบนี้มีข้อจำกัดที่ทำให้ไม่เหมาะสมกับโครงการที่มีการเปลี่ยนแปลงความต้องการบ่อยครั้งหรือต้องการการตอบสนองที่รวดเร็ว การที่ไม่สามารถย้อนกลับไปแก้ไขข้อผิดพลาดในขั้นตอนที่ผ่านมาได้อาจทำให้เกิดปัญหาที่ซับซ้อนในขั้นตอนถัดไป ดังนั้น การใช้แบบจำลองน้ำตกควรพิจารณาความเหมาะสมของโครงการก่อนที่จะนำมาใช้งานจริง แผนภาพแสดงแบบจำลองน้ำตก ดังภาพ 2.2

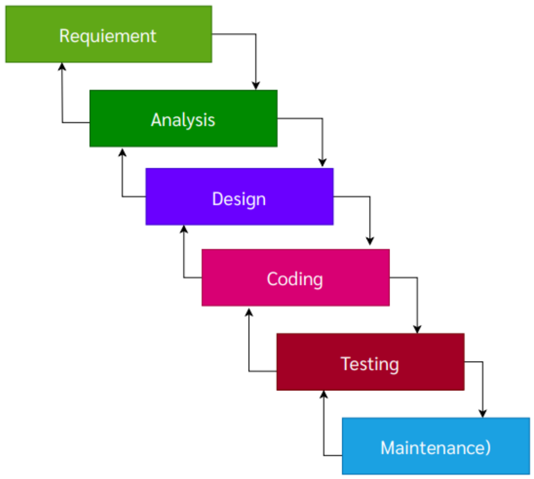


ภาพที่ 2.2 แบบจำลองน้ำตก

2. แบบจำลองปรับปรุงน้ำตก (Adapted Waterfall Model)

อรยา ปรีชาพานิช (2557) วงจรการพัฒนาระบบแบบปรับปรุงน้ำตก (Adapted Waterfall) เป็นการพัฒนาระบบที่ได้รับแรงบันดาลใจจากแบบจำลองน้ำตกแต่ได้มีการปรับแต่งเพื่อเข้ากับสภาวะแวดล้อมที่มีการเปลี่ยนแปลงต่อเนื่อง โดยในแต่ละขั้นตอนของกระบวนการพัฒนา จะมีการวิเคราะห์และทดสอบเพื่อประเมินคุณภาพและความถูกต้องของผลงาน ก่อนที่จะเข้าสู่ขั้นตอนถัดไป เป็นรูปแบบที่ช่วยให้สามารถทำการปรับปรุงและแก้ไขปัญหาที่เกิดขึ้นได้ในขั้นตอนก่อนหน้านั้น โดยไม่จำเป็นต้องย้อนกลับไปขั้นตอนเริ่มต้นใหม่

แบบวงจรนี้ช่วยให้สามารถตรวจสอบและปรับปรุงระบบในทุกขั้นตอนของกระบวนการพัฒนา ซึ่งทำให้มีโอกาสลดความเสี่ยงในการพบข้อผิดพลาดหรือปัญหาในขั้นตอนถัดไป การที่ไม่จำเป็นต้องย้อนกลับไปขั้นตอนที่ผ่านมาเพื่อแก้ไขปัญหาช่วยประหยัดเวลาและทรัพยากรที่ใช้ในการพัฒนา อย่างไรก็ตาม ความเหมาะสมของแบบวงจรนี้ยังควรถูกพิจารณาตามความซับซ้อนและความเหมาะสมของโครงการเพื่อให้สามารถนำมาใช้งานได้อย่างเต็มประสิทธิภาพ แผนภาพแสดงแบบจำลองปรับปรุงน้ำตก ดังภาพที่ 2.3



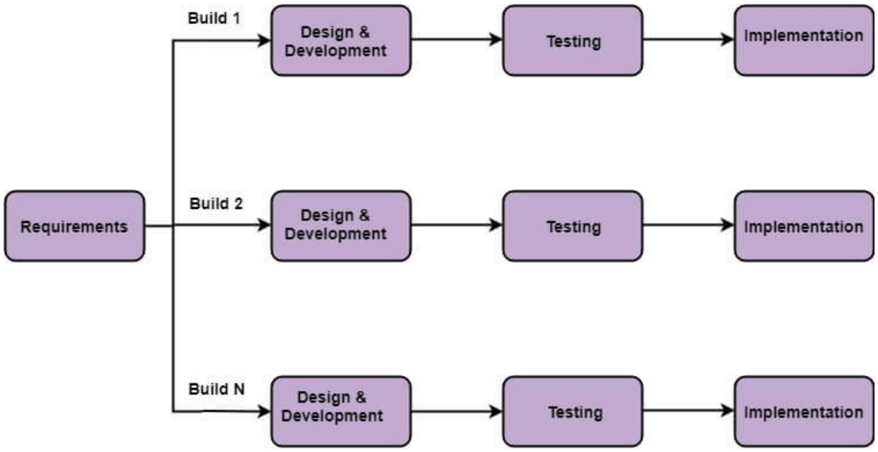
ภาพที่ 2.3 แบบจำลองปรับปรุงน้ำตกที่วนซ้ำได้

3. แบบจำลองแบบเพิ่มผลลัพธ์ (Incremental Model)

โอภาส เอี่ยมสิริวงศ์ (2556) แบบจำลองแบบเพิ่มผลลัพธ์ (Incremental Model) เป็นการพัฒนาแบบจำลองที่ได้รับแรงบันดาลใจจากแบบจำลองน้ำตก แต่ได้มีการปรับปรุงและเพิ่มประสิทธิภาพเพื่อให้การพัฒนาระบบดียิ่งขึ้น แบบจำลองน้ำตกมีข้อเสียที่กระบวนการทดสอบอยู่ในขั้นตอนท้ายของกระบวนการพัฒนา ซึ่งอาจทำให้เกิดการย้อนกลับไปเริ่มต้นโครงการใหม่ถ้าการวางแผนและการจัดการไม่เพียงพอ แบบจำลองแบบเพิ่มผลลัพธ์

จึงได้มาเพื่อลดข้อเสียดังกล่าว โดยการเพิ่มส่วนของการออกแบบและพัฒนาเป็นส่วนงานย่อยที่สามารถเพิ่มเข้าสู่ระบบได้ตามลำดับ แต่ก่อนที่จะเพิ่มเข้าสู่ระบบ แต่ละส่วนงานย่อยจะถูกทดสอบและปรับปรุงในรอบการทำงานซ้ำ (Iteration) ในรูปแบบของระบบตรวจสอบ

แบบวงจรนี้ช่วยให้การพัฒนาระบบมีความยืดหยุ่นมากขึ้นในการปรับปรุงและแก้ไข นอกจากนี้ยังช่วยลดความเสี่ยงที่จะต้องย้อนกลับไปเริ่มต้นโครงการใหม่ ด้วยการแบ่งการพัฒนาเป็นส่วนย่อยที่ทดสอบและปรับปรุงในขั้นตอนที่กำหนดไว้ล่วงหน้า อย่างไรก็ตาม ความเหมาะสมของแบบจำลองนี้ยังควรถูกพิจารณาตามความซับซ้อนและลักษณะของโครงการเพื่อให้สามารถนำมาใช้งานได้อย่างเต็มประสิทธิภาพดังภาพที่ 2.4



ภาพที่ 2.4 แบบจำลองแบบเพิ่มผลลัพธ์

4. แบบจำลองแบบเกลียว (Spiral Model)

แบบจำลองแบบเกลียว มีแต่แนวคิดที่กล่าวถึงการทำงานทุกอย่างล้วนแล้วแต่มีความเสี่ยง (Risk) ทั้งสิ้นและความเสี่ยงสามารถเกิดขึ้นได้ทุกระยะตั้งแต่เริ่มโครงการไปจนกระทั่งจบโครงการ เช่น เจ้าหน้าที่ที่รับผิดชอบหลักในโครงการนี้ได้ลาออกจากองค์กร หรือบริษัทที่ผลิตอุปกรณ์สำหรับใช้ในการโครงการได้ยกเลิกการผลิตอุปกรณ์ดังกล่าว เป็นต้น (อรยา ปรีชาพานิช, 2557)

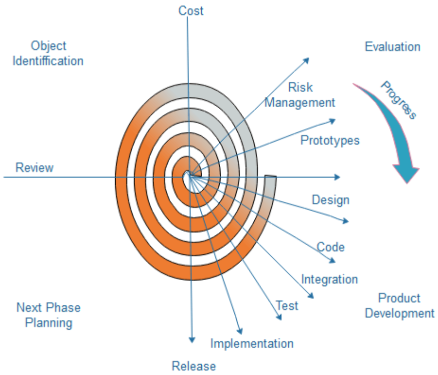
การพัฒนาระบบที่มีประสิทธิภาพจึงต้องมีการประเมินความเสี่ยงและหาวิธีการจัดการความเสี่ยงที่เหมาะสมกับโครงการเป็นระยะ ๆ ซึ่งเป็นที่มาของแบบจำลองแบบเกลียวที่มีการดำเนินการตามขั้นตอนของวงจรการพัฒนาระบบเป็นวงรอบ ประกอบด้วย

1) ขั้นตอนการวางแผนการดำเนินงานในแต่ละรอบ ในที่นี้จะรวบรวมถึงการปรับแผนให้เหมาะสมกับสถานการณ์ปัจจุบันด้วย

2) ขั้นตอนการจัดการความเสี่ยง เป็นขั้นตอนที่มีการสร้างโปรแกรมต้นแบบที่ใช้เป็นเครื่องมือในการวิเคราะห์และกำหนดแนวทางในการจัดการความเสี่ยงที่อาจเกิดขึ้นในแต่ละวงรอบ

3) ขั้นตอนการพัฒนาและทดสอบระบบเป็นการนำผลที่ได้จากขั้นตอนที่หนึ่งและที่สองมาพัฒนาระบบแต่ละเวอร์ชันจากนั้นจึงทดสอบระบบตามหลักการวิศวกรรมซอฟต์แวร์

4) ขั้นตอนการประเมินผลระบบจะทำบนผู้ที่เกี่ยวข้องทุกฝ่ายจากนั้นจึงติดตั้งเพื่อใช้งานระบบเช่นนั้น ในกรณีที่ระบบมีความครบถ้วนสมบูรณ์ตามที่กำหนดไว้ในสัญญาจ้างก็จะสามารถส่งมอบงานและ ถ้าโครงการใดแต่ถ้าระบบยังไม่ครบถ้วนสมบูรณ์ก็จะเข้าสู่ขั้นตอนการทำงานในรอบถัดไป ดังภาพที่ 2.5



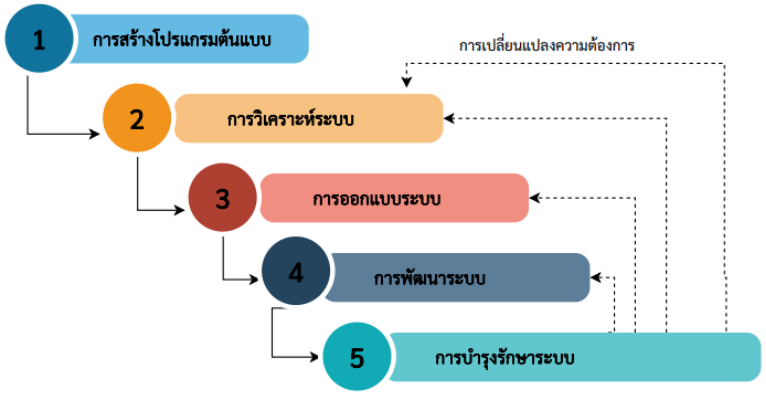
ภาพที่ 2.5 แบบจำลองแบบเกลียว

5. แบบจำลองการสร้างโปรแกรมต้นแบบในระยะเริ่มต้น (Rapid Prototyping Model)

แบบจำลองการสร้างโปรแกรมต้นแบบในระยะเริ่มต้น (Rapid Prototyping Model) ได้เจริญพัฒนามาจากแนวคิดของแบบจำลองน้ำตก แบบนี้ได้รับการปรับแต่งเพื่อให้เน้นการสร้างโปรแกรมต้นแบบในระยะเริ่มต้นของโครงการ เพื่อที่จะช่วยให้ผู้ใช้งานรู้สึกมั่นใจกับความต้องการและฟีเจอร์ของระบบก่อนการพัฒนาเต็มรูปแบบ (อรยา ปรีชาพานิช, 2557)

แบบนี้มีขั้นตอนการดำเนินงานคล้ายกับแบบจำลองน้ำตก แต่มีความแตกต่างในการเน้นการสร้างโปรแกรมต้นแบบ (Prototype) ในขั้นตอนแรกของการพัฒนา โดยการสร้างโปรแกรมต้นแบบนี้จะถูกนำไปให้ผู้ใช้งานทดสอบและให้คำติชม ซึ่งจะช่วยให้ผู้ใช้รับรู้และเข้าใจความต้องการจริง ๆ และสามารถระบุความเหมาะสมและปรับปรุงฟีเจอร์ต่าง ๆ ก่อนที่จะเข้าสู่กระบวนการพัฒนาเต็มรูปแบบ

ดังนั้น การสร้างโปรแกรมต้นแบบที่มีฟังก์ชันการทำงานและรูปแบบรายงานต่าง ๆ เทียบได้กับระบบงานที่พัฒนาจริง แต่ยังไม่มีการจัดเก็บข้อมูลและยังไม่มีการตรวจสอบความถูกต้องของข้อมูล จะช่วยให้ผู้ใช้ระบบสามารถมองเห็นภาพรวมทั้งหมดของระบบ ก่อนเริ่มต้นดำเนินการวิเคราะห์และออกแบบระบบต่อไป ดังภาพที่ 2.6



ภาพที่ 2.6 แบบจำลองการสร้างโปรแกรมต้นแบบในระยะเริ่มต้น

ข้อควรระวังของการใช้แบบจำลองที่มีการสร้างโปรแกรมต้นแบบในระยะเริ่มต้นของวงจรการพัฒนาระบบ คือ ทีมนักพัฒนาระบบจะต้องตระหนักถึงที่มาของโปรแกรมต้นแบบว่าได้สร้างขึ้นอย่างรวดเร็วและผ่านการปรับปรุงอย่างต่อเนื่องเพื่อให้สามารถสรุปความต้องการใช้งานระบบที่แท้จริงของผู้ใช้ทุกฝ่ายที่เกี่ยวข้อง ดังนั้น เมื่อบรรลุวัตถุประสงค์แล้วจึงยกเลิกการใช้โปรแกรมต้นแบบนั้นอย่างเด็ดขาด และพัฒนาระบบขึ้นมาใหม่ตามความต้องการที่แท้จริงของผู้ใช้งานตามหลักการวิศวกรรมซอฟต์แวร์

ระเบียบวิธีการพัฒนาระบบแบบดั้งเดิม (Traditional System Development Methodology)

ระเบียบวิธีการพัฒนาระบบ หมายถึง กระบวนการดำเนินงานตามขั้นตอนที่ถูกกำหนดไว้ในวงจรการพัฒนาระบบ เพื่อให้ผลลัพธ์ที่ได้เป็นระบบที่มีประสิทธิภาพและสามารถทำงานได้อย่างมีประสิทธิผล ในปัจจุบันนี้มีระเบียบวิธีการพัฒนาระบบหลายแนวทางที่สามารถเลือกใช้ได้ตามความเหมาะสมของโครงการ และมีการพิจารณาปัจจัยที่เกี่ยวข้อง เช่น ขนาดและความซับซ้อนของระบบที่กำลังพัฒนา ความสัมพันธ์ของระบบที่กำลังพัฒนาและระบบอื่น ๆ ในองค์กร สภาพแวดล้อมขององค์กร วัฒนธรรมขององค์กร รวมถึงทรัพยากรที่ใช้ในการดำเนินโครงการ เป็นต้น

การเลือกใช้ระเบียบวิธีการพัฒนาระบบที่เหมาะสมนั้นเกิดขึ้นจากการพิจารณาความเหมาะสมของแต่ละแนวทาง และการให้คำนึงถึงปัจจัยที่อาจมีผลต่อการพัฒนาระบบ เช่น ระดับความซับซ้อนและขนาดของระบบ ความสัมพันธ์ระหว่างระบบที่กำลังพัฒนาและระบบอื่น ๆ ในองค์กร รวมถึงปัจจัยทางองค์กรและการบริหารทรัพยากร การเลือกใช้ระเบียบวิธีการพัฒนาระบบที่เหมาะสมนี้ช่วยให้การพัฒนาระบบเป็นไปได้อย่างมี

ประสิทธิผล โดยคำนึงถึงความเฉพาะเจาะจงของโครงการและสภาพแวดล้อมขององค์กร อย่างไรก็ตาม ความเหมาะสมของระเบียบวิธีการนี้ยังควรพิจารณาตามลักษณะและความซับซ้อนของโครงการเพื่อให้สามารถนำมาใช้งานได้อย่างเต็มประสิทธิภาพ ในหัวข้อนี้จะอธิบายถึงองค์ประกอบของระบบเบียบวิธีการพัฒนาระบบ และระเบียบวิธีการพัฒนาระบบ 6 ระเบียบวิธี ดังต่อไปนี้

1. องค์ประกอบของระเบียบวิธีการพัฒนาระบบ

ระเบียบวิธีการพัฒนาระบบประกอบด้วย 3 ส่วน คือ แบบจำลอง (Model) เครื่องมือ (Tools) และเทคนิค (Techniques) ที่มีความสัมพันธ์กันเพื่อสนับสนุนวิธีการดำเนินงานที่แตกต่างกันไป (อรยา ปรีชาพานิช, 2557)

1) แบบจำลอง (Models)

แบบจำลองเป็นแผนภาพที่ใช้สัญลักษณ์แทนข้อความ หรือคำอธิบาย เน้นการสื่อสารให้เห็นถึงภาพรวมทั้งระบบ โดยแบบจำลองที่สามารถประยุกต์ใช้ในการพัฒนาระบบ ได้แก่

1.1) แบบจำลองกระบวนการพัฒนาระบบ (System Development Process Model)

1.2) ผังงานโปรแกรม (Program Flowchart)

1.3) แผนภาพกระแสงาน (Workflow Diagram)

1.4) แผนภาพกระแสข้อมูล (Data Flow Diagram)

1.5) แผนภาพแสดงความสัมพันธ์ของข้อมูล (Entity Relationship Diagram)

1.6) แผนภาพยูสเคส (Use Case Diagram)

1.7) แผนภาพคลาส (Class Diagram)

1.8) แผนภาพแกนต์ (Gantt Chart)

2) เครื่องมือ (Tools

เครื่องมือที่ใช้ในการพัฒนาระบบ หมายถึง ซอฟต์แวร์ที่ช่วยในการสร้างและตรวจสอบความถูกต้องของแบบจำลอง ช่วยสร้างรายงานและแบบฟอร์มต่าง ๆ รวมไปถึงการสร้างโปรแกรมโดยอัตโนมัติ ได้แก่

2.1) เครื่องมือวาดแผนภาพ (DrawingTools)

2.2) เครื่องมือจัดการโครงการ (Project Management Tools)

2.3) เครื่องมือจัดการฐานข้อมูล (DatabaseManagement Tools)

2.4) เครื่องมือทางวิศวกรรมซอฟต์แวร์ที่ใช้คอมพิวเตอร์ช่วยดำเนินการ (Computer - Aided Software Engineering Tools: CASE Tools)

2.5) เครื่องมือสร้างรหัสโปรแกรม (Code Generator Tools)

2.6) เครื่องมือสร้างรายงาน (Report Generator Tools)

3) เทคนิค (Techniques)

เทคนิคในการพัฒนาระบบ หมายถึง แนวทางที่ช่วยแนะนำกระบวนการเพื่อให้นักวิเคราะห์ระบบสามารถนำเทคนิคไปดำเนินกิจกรรมการพัฒนาระบบเพื่อให้เกิดความสมบูรณ์ยิ่งขึ้น ตัวอย่างเทคนิคต่าง ๆ ได้แก่

3.1) เทคนิคในการสัมภาษณ์ (Interview Technique)

3.2) เทคนิดในการจัดการโครงการ (Project Management Technique)

3.3) เทคนิคในการวิเคราะห์ระบบเชิงโครงสร้าง (Structure Analysis Technique)

3.4) เทคนิคในการทดสอบซอฟต์แวร์ (Software Testing Technique)

ในปัจจุบันได้มีระเบียบวิธีในการพัฒนาระบบหลายแนวทาง ซึ่งนักวิเคราะห์ระบบสามารถเลือกใช้ได้ตามความเหมาะสม ซึ่งในบทนี้จะยกตัวอย่างระเบียบวิธีในการพัฒนาระบบในหัวข้อถัดไป

2. ระเบียบวิธีการพัฒนาระบบเชิงโครงสร้าง (Structure Analysis and Design: SAD)

ระเบียบวิธีการพัฒนาระบบเชิงโครงสร้างเป็นกระบวนการที่ใช้ในการพัฒนาระบบที่มีลำดับขั้นตอนที่ถูกกำหนดไว้เพื่อให้ผลลัพธ์ที่สมบูรณ์และถูกต้อง แต่ละขั้นตอนในกระบวนการนี้ถูกออกแบบให้เชื่อมโยงกันอย่างเสริมสร้าง ซึ่งความเชื่อมโยงนี้ส่งผลให้ความถูกต้องและความสมบูรณ์ของข้อมูลและผลลัพธ์ที่เกิดขึ้นในขั้นตอนก่อนหน้านั้นมีผลต่อขั้นตอนถัดไป โดยแบบจำลองน้ำตกเป็นตัวอย่างที่นำเสนอความหมายดังกล่าวในหัวข้อก่อนหน้านี้

คุณสมบัติที่ดีของระเบียบวิธีการพัฒนาระบบเชิงโครงสร้าง คือ เน้นที่แต่ละขั้นตอนจะต้องดำเนินการอย่างรอบคอบและเสร็จสิ้นตามลำดับ ซึ่งช่วยให้ได้ผลลัพธ์ที่เป็นไปตามเป้าหมายและมีคุณภาพสูงสุด การพิจารณาผลและการแก้ไขข้อผิดพลาดจะเกิดขึ้นในขั้นตอนที่มาก่อนหน้านั้น จึงส่งผลให้การพัฒนาระบบดำเนินไปได้อย่างมีความแม่นยำ

ข้อดีของระเบียบวิธีการพัฒนาระบบเชิงโครงสร้างจะทำงานในแต่ละขั้นตอนของวงจรการพัฒนาระบบอย่างรอบคอบเพื่อให้ได้ผลลัพธ์ที่ถูกต้องมากที่สุด และจะต้องผ่านการตรวจสอบกับผู้ใช้ทุกฝ่ายที่เกี่ยวข้องก่อนดำเนินการในขั้นตอนถัดไป

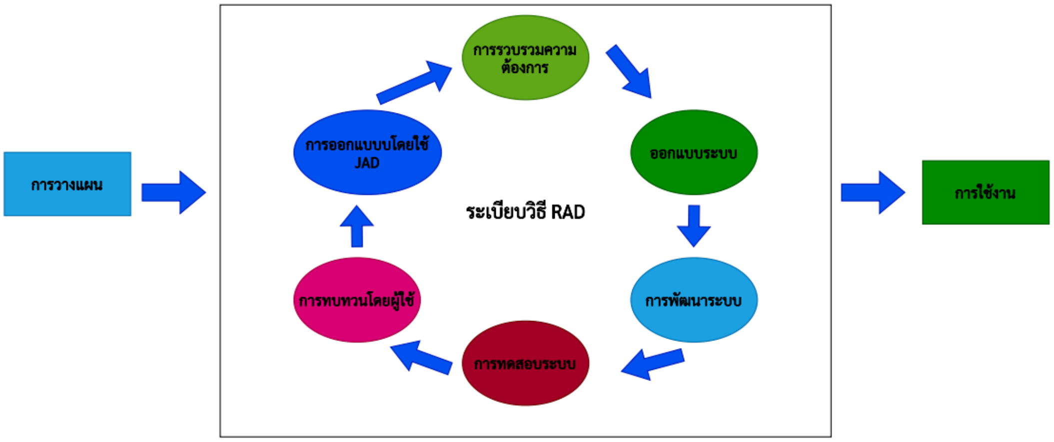
ข้อจำกัดของระเบียบวิธีการพัฒนาระบบเชิงโครงสร้างจะใช้ระยะเวลาในการดำเนินงานตั้งแต่เริ่มต้นจนกระทั่งสิ้นสุดโครงการค่อนข้างนาน บางครั้งความต้องการใช้งาน หรือเทคโนโลยีสารสนเทศที่สามารถประยุกต์ใช้งานกับระบบ อาจเปลี่ยนแปลงไปจากที่ได้กำหนดไว้ในระยะเริ่มต้นโครงการ

3. ระเบียบวิธีการพัฒนาโปรแกรมประยุกต์แบบรวดเร็ว (Rapid Application Development: RAD)

ระเบียบวิธีการพัฒนาโปรแกรมประยุกต์แบบรวดเร็ว (Rapid Application Development: RAD) เป็นกระบวนการที่ใช้ในการพัฒนาโปรแกรมประยุกต์หรือแอปพลิเคชันโดยเน้นความเร็วในการพัฒนา และการนำระบบไปใช้งานจริงเป็นหลัก วิธีการนี้รักษาคุณภาพของระบบอย่างมาก พร้อมทั้งลดระยะเวลาและค่าใช้จ่ายในการดำเนินโครงการ เมื่อเปรียบเทียบกับการใช้ระเบียบวิธีการพัฒนาระบบเชิงโครงสร้างแบบดั้งเดิม ในกระบวนการนี้มักมีการใช้เครื่องมือสนับสนุน เช่น CASE TOOL เพื่อช่วยในกระบวนการพัฒนา สิ่งนี้ทำให้ระบบที่ถูกพัฒนาด้วยวิธีการนี้สามารถพัฒนาในระยะเวลาที่สั้นได้โดยคงคุณภาพของผลลัพธ์

วิธีการนี้ช่วยลดระยะเวลาในแต่ละขั้นตอนของวงจรการพัฒนาระบบ เช่น การวิเคราะห์ การออกแบบ การพัฒนา และการทดสอบ ทำให้มีความเร็วในการพัฒนาโดยรวม อย่างไรก็ตาม วิธีการนี้ยังมีการคำนึงถึงคุณภาพของผลลัพธ์ และต้องผ่านการตรวจสอบจากผู้ใช้ทุกฝ่ายที่เกี่ยวข้องก่อนดำเนินการขั้นตอนถัดไป

อย่างไรก็ตาม ระเบียบวิธีการพัฒนาโปรแกรมประยุกต์แบบรวดเร็วก็มีข้อจำกัด ที่เกี่ยวข้องกับการใช้เวลาในการพัฒนา ซึ่งอาจทำให้ความต้องการใช้งานหรือเทคโนโลยีที่เป็นที่ประยุกต์ใช้กับระบบเปลี่ยนแปลงไปจากที่กำหนดไว้ในระยะเริ่มต้นของโครงการ ดังแสดงในภาพที่ 2.7



ภาพที่ 2.7 ระเบียบวิธีการพัฒนาแอปพลิเคชันแบบรวดเร็ว

วิธีการพัฒนาโปรแกรมประยุกต์แบบรวดเร็วเป็นวงจรการพัฒนาระบบที่ใช้ระยะเวลาในการพัฒนารวดเร็วกว่า และคุณภาพดีกว่าวิธีพัฒนาระบบงานแบบดั้งเดิม โดยนำเครื่องมือซอฟต์แวร์มาช่วยในการพัฒนาระบบ

1) ขั้นตอนการพัฒนาโปรแกรมประยุกต์แบบรวดเร็วมี 4 ขั้นตอน ดังนี้

1.1) การกำหนดความต้องการ ในขั้นตอนนี้ผู้ใช้และผู้บริหารที่เกี่ยวข้องจะทำการกำหนดว่าระบบควรจะมีหน้าที่และมีงานใดบ้าง การกำหนดความต้องการนี้อาจใช้วิธีสัมมนาเชิงปฏิบัติการ โดยการนำผู้ใช้ ผู้บริหารและผู้ที่เกี่ยวข้องเข้าร่วมสัมมนา ในระหว่างการสัมมนาอาจมีการพิจารณาถึงปัญหาต่าง ๆ และมีบุคลากรด้านเทคโนโลยีสารสนเทศช่วยชี้แนะว่าจะสามารถนำคอมพิวเตอร์มาช่วยในการแก้ปัญหาได้อย่างไร

1.2) การออกแบบโดยผู้ใช้ ในขั้นตอนของการออกแบบนั้นผู้ใช้จะมีส่วนร่วมอย่างมากในการออกแบบ เช่น การออกแบบหน้าจอ การออกแบบรูปแบบรายงาน เป็นต้น โดยอาจใช้สัมมนาเชิงปฏิบัติการเช่นเดียวกับในขั้นตอนการกำหนดความต้องการ และมีการใช้เครื่องมือซอฟต์แวร์มาช่วยในการออกแบบระบบ ทำระบบต้นแบบ ปรับปรุงต้นแบบตามความต้องการ ซึ่งเครื่องมือซอฟต์แวร์นี้สามารถช่วยอำนวยความสะดวกในการปรับปรุงต้นแบบและช่วยตรวจสอบความถูกต้องของต้นแบบ

1.3) การพัฒนาระบบ เครื่องมือซอฟต์แวร์ถูกนำมาช่วยในการพัฒนาระบบ เช่น สร้างส่วนติดต่อกับผู้ใช้แบบกราฟฟิกสั่งโปรแกรมที่สามารถนำไปใช้งานได้อย่างรวดเร็ว โดยในระหว่างการสร้างจะทำการทดสอบไปด้วย ในขั้นตอนนี้ต้องอาศัยความร่วมมือระหว่างผู้ใช้นักพัฒนาระบบ หากผู้ใช้ต้องการแก้ไขปรับเปลี่ยน เช่น รายการข้อมูลที่ปรากฏหน้าจอไม่ครบถ้วน ก็จะแจ้งให้ผู้พัฒนาระบบทำการปรับเปลี่ยนโปรแกรมซึ่งสามารถทำได้ง่ายใช้เวลาไม่นาน

1.4) การเปลี่ยนระบบ เป็นการเปลี่ยนจากระบบเดิมมาเป็นระบบใหม่ ซึ่งจะมีขั้นตอนเหมือนกับวิธีการพัฒนางานแบบดั้งเดิม เช่น การทดสอบ การฝึกอบรมผู้ใช้ และการใช้ระบบใหม่

2) เทคนิคสำคัญของการพัฒนาระบบตามวิธีการพัฒนาโปรแกรมประยุกต์แบบรวดเร็ว

2.1) พัฒนาต้นแบบได้อย่างรวดเร็ว

2.2) เป็นแหล่งรวมเครื่องมือเพื่อพัฒนาระบบงานประยุกต์

2.3) มีทีมงานที่เชี่ยวชาญการใช้เครื่องมือ

2.4) เป็นแนวร่วมปฏิบัติการกับการพัฒนาระบบประยุกต์แบบร่วม

2.5) มีกรอบระยะเวลาการพัฒนาระบบที่จำกัด

3) ข้อดีของวิธีการพัฒนาโปรแกรมประยุกต์แบบรวดเร็ว (อรยา ปรีชาพานิช, 2557)

3.1) สามารถลดระยะเวลา ค่าใช้จ่าย และทรัพยากรต่าง ๆ ในการดำเนินงานโครงการได้

3.2) ลดความเสี่ยงจากข้อผิดพลาดในการรวบรวมข้อมูลความต้องการ หรือความต้องการที่เปลี่ยนแปลงไปของผู้ใช้ เนื่องจากมีการใช้วิธีการพัฒนาระบบแบบร่วมอย่างต่อเนื่องทั้งโครงการ

3.3) สามารถสร้างความรู้สึกเป็นเจ้าของร่วมกันของผู้ที่มีส่วนเกี่ยวข้องกับระบบทุกฝ่าย

3.4) สามารถพัฒนาระบบตรงกับความต้องการใช้งานของผู้ใช้มากยิ่งขึ้น เนื่องจากผู้ใช้สามารถตรวจสอบจากโปรแกรมต้นแบบที่สร้างขึ้นก่อนดำเนินการพัฒนาระบบจริง

4) ข้อจำกัดของการพัฒนาระบบงานประยุกต์แบบรวดเร็ว (อรยา ปรีชาพานิช, 2557)

4.1) ขาดความยืดหยุ่นในการปรับขนาดของระบบ (System Scalability) เนื่องจากการพัฒนาระบบจะอ้างอิงจากโปรแกรมต้นแบบที่ผ่านการพิจารณาแล้วเป็นหลัก

4.2) ทีมงานพัฒนาระบบอาจละเลยในส่วนของการวิเคราะห์สภาพแวดล้อมทางธุรกิจ และการเปลี่ยนแปลงที่อาจส่งผลกระทบกับระบบในอนาคต เนื่องจากคาดหวังว่า JAD จะช่วยในการระดมสมองร่วมกันระหว่างผู้ที่มีส่วนเกี่ยวข้องกับระบบทุกฝ่ายอยู่แล้ว

4.3) การใช้วิธีการพัฒนาแบบร่วมจะต้องพิจารณาถึงวัฒนธรรมขององค์กร และการให้ความร่วมมือของผู้ที่มีส่วนเกี่ยวข้องกับระบบในภาพรวม มิเช่นนั้น อาจเกิดความล่าช้าในการดำเนินงาน และอาจสร้างสัมพันธภาพที่ไม่ดีต่อกันระหว่างทีมงานพัฒนาระบบและองค์กรได้

4.4) คุณภาพของเอกสารที่เกี่ยวข้องกับการพัฒนาระบบอาจลดลงได้ เนื่องจากทีมงานพัฒนาระบบพยายามที่จะลดระยะเวลาของวงจรการพัฒนาระบบให้สั้นลงเพื่อส่งมอบงานในเวลาที่รวดเร็วยิ่งขึ้น

4. ระเบียบวิธีการพัฒนาโปรแกรมประยุกต์แบบร่วมกัน (Joint Application Development: JAD)

ระเบียบวิธีการพัฒนาโปรแกรมประยุกต์แบบร่วมกัน (Joint Application Development: JAD) เป็นกระบวนการพัฒนาระบบออกแบบที่เน้นการร่วมมือระหว่างบุคคลในองค์กรและผู้เชี่ยวชาญทางเทคโนโลยีสารสนเทศ ผ่านการจัดประชุมเชิงปฏิบัติการ (Workshop) การพัฒนาร่วมกันนี้เกิดขึ้นเพื่อพัฒนาระบบที่มีคุณภาพและสมบูรณ์ในระยะเวลาที่สั้น และส่งมอบตามระยะเวลาที่กำหนดไว้ได้

วิธีการนี้ถูกพัฒนาโดยบริษัทไอบีเอ็ม (IBM) ในปีค.ศ. 1970 และเป็นวิธีการที่มีประโยชน์มากในการรวบรวมความต้องการจากผู้ใช้ รูปแบบการทำงานของ JAD นั้นเป็นการรวมกลุ่มบุคคลที่เกี่ยวข้องในการพัฒนาระบบ และใช้พื้นที่ในสำนักงานเป็นห้องปฏิบัติการหรือศูนย์การทำงานร่วมกัน ทุกคนมีเป้าหมายและขอบเขตงานที่ชัดเจน และมุ่งหวังในการหารือและพัฒนาระบบอย่างมีประสิทธิผล อย่างไรก็ตาม วิธีการนี้ไม่ใช่การประชุมแบบธรรมดา เนื่องจากมีการจัดกลุ่มทำงานและกำหนดเป้าหมายอย่างชัดเจนเพื่อประสบความสำเร็จในการพัฒนา

โครงการวิธีการพัฒนาโปรแกรมประยุกต์แบบร่วมกัน ประกอบด้วย

1) การสนทนาร่วมกัน มุ่งเน้นความคล่องแคล่วและสะดวกภายใต้กฎเกณฑ์ที่บังคับใช้

2) มีกลุ่มผู้ใช้ประมาณ 3 - 5 คนที่พร้อมเข้าร่วมประชุมเชิงปฏิบัติการ

3) นักพัฒนาที่มีความเชี่ยวชาญด้านเทคโนโลยีสารสนเทศประมาณ 2 - 3 คน

4) ผู้ที่มีอำนาจซึ่งอาจเป็นผู้จัดการอาวุโสที่สามารถตัดสินและชี้ขาดได้ ในกรณีเกิดข้อขัดแย้งระหว่างกันที่ไม่สามารถตกลงกันได้

5) ผู้สังเกตการณ์ประมาณ 2 - 3 คน ซึ่งมีหน้าที่สังเกตการณ์เท่านั้น ไม่ต้องวิจารณ์หรือให้ความคิดเห็น

6) ผู้เชี่ยวชาญในการสรุปเนื้อหาสาระสำคัญ ที่มีความเข้าใจในระบบงานธุรกิจและเทคโนโลยีสารสนเทศ สามารถสรุปเนื้อหาสาระสำคัญได้กระชับและชัดเจน

จะเห็นได้ว่าวิธีการพัฒนาระบบด้วยวิธีการนี้ไม่สามารถทำงานได้ด้วยตัวคนเดียว ทุกคนในทีมงานจะต้องร่วมกันพัฒนาและมีจุดมุ่งหมายเดียวกัน เพื่อพัฒนาระบบให้สมบูรณ์และใช้เวลาอันสั้น ระบบงานที่พัฒนาด้วยวิธีการนี้มีความสมบูรณ์ของโครงการค่อนข้างสูง เนื่องจากมีผู้เชี่ยวชาญด้านต่าง ๆ พร้อมที่จะมุ่งมั่นทำงานให้สำเร็จรวมทั้งผู้ใช้งานที่มีส่วนร่วมในการเข้าประชุมเชิงปฏิบัติการร่วมกันทีมงานตลอดโครงการ

5. ระเบียบวิธีการพัฒนาระบบแบบ Rational Unified Process (RUP)

ระเบียบวิธีการพัฒนาระบบแบบ Rational Unified Process เป็นวิธีการพัฒนาระบบที่อาศัยหลักการของการทำงานแบบวนซ้ำเพื่อเพิ่มผลลัพธ์ และเป็นระเบียบวิธีการพัฒนาระบบเชิงวัตถุ (Object Oriented System Development Methodology) เป็นกระบวนการที่พัฒนาขึ้นโดยบริษัท Rational Software ซึ่งรู้จักในนามของผลิตภัณฑ์ UML (Unified Modeling Language) จุดประสงค์ของวิธีการนี้ต้องการให้ทีมงานพัฒนาระบบที่มีคุณภาพสูงตรงตามความต้องการของผู้ใช้ ภายใต้งบประมาณและระยะเวลาที่กำหนดไว้ โดยพื้นฐานสำคัญของวิธีการนี้คือ การสร้างแบบจำลองและการจัดการแบบจำลองด้วยภาษายูเอ็มแอล UML โดยแบ่งระยะการดำเนินงานเป็น 4 ระยะ (อรยา ปรีชาพานิช, 2557) คือ

1) ระยะเริ่มต้น (Inception Phase) เป็นการทำความเข้าใจถึงระบบที่จะพัฒนาในภาพรวมทั้งหมดรวมไปถึงค่าใช้จ่าย ระยะเวลาการดำเนินงาน และความเสี่ยงที่อาจเกิดขึ้นกับโครงการ เพื่อกำหนดวิธีการดำเนินงานและเครื่องมือที่จำเป็นต้องใช้ในการพัฒนาระบบ

2) ระยะการกำหนดรายละเอียด (Elaboration Phase) เป็นการกำหนดแผนการดำเนินงานโครงการอย่างละเอียด รวมไปถึงการวิเคราะห์และออกแบบเพื่อกำหนดสถาปัตยกรรมของระบบ และโครงสร้างฐานข้อมูล ในขั้นตอนนี้สามารถทำซ้ำได้หลายรอบขึ้นอยู่กับขนาดความซับซ้อน และความเสี่ยงของระบบที่ได้ประเมินไว้ในระยะเริ่มต้น

3) ระยะการพัฒนาระบบ (Instruction Phase) เป็นการเขียนโปรแกรม และการทดสอบระบบตามหลักการวิศวกรรมซอฟต์แวร์ โดยทั่วไปการดำเนินงานระยะนี้จะใช้เวลายาวนานที่สุดเมื่อเทียบกับระยะอื่น ๆ เพื่อให้ได้ระบบที่มีคุณภาพ ซึ่งในขั้นตอนนี้สามารถทำซ้ำได้หลายรอบขึ้นอยู่กับความถูกต้อง ความซับซ้อน และความเสี่ยงที่เกิดจากการวิเคราะห์และออกแบบระบบจากระยะการกำหนดรายละเอียด

4) ระยะการนำไปใช้งานจริง (Transition Phase) เป็นการติดตั้ง ฝึกอบรมการใช้งาน และใช้งานระบบจริง ซึ่งในแต่ละระยะจะมีกิจกรรมต่าง ๆ เกิดขึ้นตามขั้นตอนของวงจรการพัฒนาระบบ ประกอบด้วย การสร้างแบบจำลองทางธุรกิจ การรวบรวมความต้องการจากผู้ใช้งานการวิเคราะห์และออกแบบระบบ การพัฒนาระบบ การทดสอบระบบ และการติดตั้งใช้งานระบบ นอกจากนั้นวิธีการนี้ยังได้เพิ่มมุมมองของการจัดการในส่วนของโครงแบบและการเปลี่ยนแปลงความต้องการใช้งานระบบ การจัดการโครงการ และการจัดการสภาพแวดล้อมในทุก ๆ ระยะของการดำเนินงานอีกด้วย

6. ระเบียบวิธีการพัฒนาระบบแบบเชิงวัตถุ (Object-Oriented Methodology)

ระเบียบวิธีการพัฒนาระบบและเขียนโปรแกรมที่ผ่านมานิยมใช้แนวคิดเชิงโครงสร้าง (Structure) ซึ่งไม่รองรับการพัฒนาระบบขนาดใหญ่ที่มีความซับซ้อน และขาดความต่อเนื่องในขั้นตอนการพัฒนาระบบ ประกอบกับผู้ใช้และองค์กรต้องการให้เวลาในการพัฒนาระบบสั้นลง รวมถึงสมรรถนะของเครื่องคอมพิวเตอร์สูงขึ้น แนวโน้มในการพัฒนาระบบจึงมุ่งไปที่การใช้กรอบความคิดเชิงวัตถุ (Object-Oriented Paradigm)

การพัฒนาระบบแบบวัตถุเป็นการพิจารณาระบบเป็นกลุ่มของวัตถุ (Class of Object) ซึ่งทำงานร่วมกันมีการจัดกลุ่มของข้อมูลและพฤติกรรมหรือฟังก์ชั่นที่กระทำกับข้อมูลนั้นเป็นกลุ่ม ๆ ในรูปแบบของออบเจ็กต์ เนื่องจากออบเจ็กต์มีคุณสมบัติในการนำกลับมาใช้ใหม่ได้ (Reusability) การพัฒนาโปรแกรมแบบวัตถุจึงใช้เวลาในการพัฒนาน้อยกว่าวิธีอื่น

การพัฒนาระบบแบบวัตถุในยุคแรกมีความสับสนในเรื่องหลักวิธีในการพัฒนาระบบ เนื่องจากผู้นำด้านการวิเคราะห์และออกแบบระบบแบบวัตถุ ต่างคิดวิธีการและใช้แผนภาพหรือสัญลักษณ์ที่แตกต่างและหลากหลาย ทำให้ผู้พัฒนาระบบเกิดความสับสนในการเลือกวิธีที่จะนำไปใช้ และยากต่อการทำความเข้าใจระหว่างพัฒนาระบบของแต่ละโครงการ ดังนั้นในปี 1995 Grady Booch, James Rumbaugh, Ivar Jacobson ซึ่งเป็นผู้นำทางด้าน

การวิเคราะห์และออกแบบระบบเชิงวัตถุ ได้นำผลงานมาผสมผสานโดยรวมเอาข้อดีของวิธีการของแต่ละคนพัฒนามาเป็นภาษาสำหรับการจำลองภาพการวิเคราะห์และออกแบบเชิงวัตถุที่เรียกว่า ยูเอ็มแอล (Unified Model Language : UML)

UML เป็นภาษามาตรฐานสำหรับการจำลองระบบที่ได้รับความยอมรับอย่างกว้างขวาง ผู้พัฒนาระบบสามารถใช้ไดอะแกรมต่าง ๆ ของ UML เพื่อจำลองภาพ (Visualizing) ของระบบด้วยมุมมองต่างๆ เช่น จำลองภาพการกำหนดความต้องการ (Specifying) การดำเนินการสร้าง (Construction) และการบันทึก (Document) ผลงานตามขั้นตอนต่าง ๆ ในการพัฒนาระบบ

วิธีการพัฒนาระบบเชิงวัตถุจะประกอบไปด้วย 3 ขั้นตอนด้วยกัน คือ ขั้นตอนที่หนึ่ง คือ การวิเคราะห์ระบบด้วยวิธีเชิงวัตถุ (Object-Oriented Analysis: OOA) ซึ่งเป็นการกำหนดวัตถุหรือออกเจ็ตต์ต่าง ๆ ที่ต้องมีในระบบ และแสดงถึงออบเจ็กต์แต่ละตัวจะโต้ตอบกันอย่างไร เพื่อให้งานใดงานหนึ่งเสร็จสมบูรณ์ ขั้นตอนที่สอง คือ การออกแบบระบบด้วยวิธีเชิงวัตถุ (Object-Oriented Design: OOD) โดยหลังจากที่ได้ทำการวิเคราะห์และได้แบบจำลองเชิงวัตถุที่สามารถนำมาใช้เป็นหลักในการออกแบบ จากนั้นก็ดำเนินการกำหนดชนิดของออบเจ็กต์เพิ่มเติมที่มีส่วนสำคัญในการสื่อสารกับมนุษย์และอุปกรณ์ในระบบ และการแก้ไขปรับปรุงให้ดีขึ้น เพื่อจะได้นำแบบจำลองที่ออกแบบมาเป็นแบบพิมพ์เขียว (Blueprint) ที่สามารถนำไปกำหนดรายละเอียดของภาษาเพื่อการโปรแกรมภาษาเชิงวัตถุต่อไป และขั้นตอนสุดท้าย คือ การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming: OOD) ซึ่งเป็นการเขียนชุดคำสั่งโปรแกรมที่เขียนขึ้นนั้นจะถูกจัดให้มีโครงสร้างการทำงานร่วมกันของวัตถุต่างๆ

ข้อดีของวิธีการพัฒนาระบบแบบเชิงวัตถุ

1) ช่วยลดความซับซ้อนในการพัฒนาระบบ เนื่องจากสามารถแบ่งปัญหาให้เป็นออบเจ็กต์ย่อยได้ จึงสามารถจัดการกับโปรแกรมขนาดใหญ่ได้ดีกว่าวิธีเชิงโครงสร้าง

2) สามารถนำโปรแกรมที่พัฒนาแล้วกลับมาใช้ใหม่ได้

3) ช่วยให้การบำรุงรักษาระบบทำได้ง่ายขึ้น เนื่องจากการจัดเก็บข้อมูลและการดำเนินงานแยกออกจากส่วนของการเรียกใช้งานเป็นการซ่อนรายละเอียดภายในออกจากการเรียกใช้งานภายนอก ซึ่งเป็นประโยชน์ของการขยายขีดความสามารถของระบบได้ โดยไม่ต้องรื้อระบบมาก

ข้อจำกัดของวิธีการพัฒนาระบบแบบเชิงวัตถุ เนื่องจากเป็นแนวคิดและการพัฒนาเชิงวัตถุจึงเป็นสิ่งที่ยากต่อการทำความเข้าใจ และถือเป็นอุปสรรคต่อผู้ที่เริ่มต้นศึกษา และทีมนักพัฒนาระบบทั้งหมดต้องมีความเชี่ยวชาญในขั้นตอนการวิเคราะห์ระบบ ออกแบบระบบ และพัฒนาระบบ เป็นแบบเชิงวัตถุทุกขั้นตอน

ระเบียบวิธีการพัฒนาซอฟต์แวร์แบบเอจายล์ (Agile Software Development Methodology)

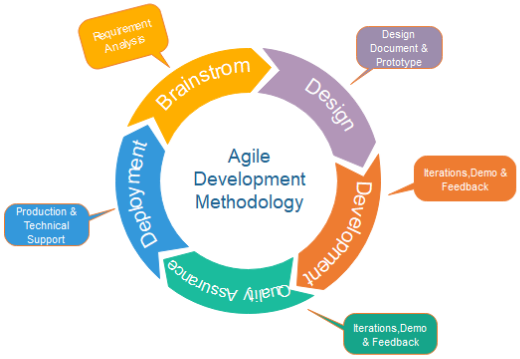
การพัฒนาซอฟต์แวร์แบบเอจายล์ (Agile Software Development) (Mason-Jones, et al.,2000) เป็นการพัฒนาซอฟต์แวร์ที่เน้นการลดระยะเวลาการทำงาน เน้นผลงานที่จับต้องได้ และให้ความสำคัญของคนและการสื่อสารภายในทีม มากกว่ามุ่งเน้นที่เอกสาร ขั้นตอนพัฒนาการประมวลผล ไม่ใช่การทำในครั้งเดียวเพื่อพัฒนาซอฟต์แวร์ ในปี ค.ศ. 2001 บุคคลที่มีชื่อเสียง 17 คน ในสาขาของการพัฒนาแบบเอจายล์ (ต่อมาเรียก Light Weight Methodologies) ได้ปรึกษากันเพื่อหาแนวทางในการสร้างซอฟต์แวร์ที่ใช้งานง่ายขึ้น เร็วขึ้น และเป็นกลาง ซึ่งเป็นนิยามที่ได้รับการยอมรับและได้รับความสนใจอย่างกว้างขวาง การพัฒนาแบบเอจายล์เป็นแนวคิดใหม่สำหรับการพัฒนาซอฟต์แวร์ ที่พยายามที่จะแทรกตัวเข้าไปในวิธีการแบบเดิม โดยใช้แบบจำลองการพัฒนาแบบ Waterfall เพื่อให้งานสั้นลง มีการแบ่งแยกงานออกเป็นส่วน ๆ เพื่อให้ทีมงานได้ทำงานอย่างมีประสิทธิภาพ ซึ่งเหมาะสมกับการตอบสนองและหาผลกำไรสูงสุดจากความต้องการของลูกค้าที่ไม่แน่นอน

การพัฒนาแบบเอจายล์ มุ่งเน้นความสำคัญที่ตัวซอฟต์แวร์ที่จะพัฒนามากกว่าการเน้นไปที่กระบวนการออกแบบและการใช้เอกสารควบคุมกระบวนการทั้งหมด ทั้งนี้ เพื่ออำนวยความสะดวกให้ทีมงานสามารถพัฒนาซอฟต์แวร์ได้รวดเร็วยิ่งขึ้น อย่างไรก็ตาม วิธีการแบบเอจายล์ยังคงอยู่บนพื้นฐานของแบบจำลองวงจร การพัฒนาซอฟต์แวร์ (Software Life Cycle Model) ซึ่งเป็นวงจรอย่างหนึ่งที่ประกอบด้วยชุดขั้นตอนหลักสำคัญ ๆ ได้แก่ การวิเคราะห์ความต้องการ การออกแบบระบบ การพัฒนาระบบและการประเมินระบบ โดยมีกระบวนการทำกิจกรรมวนซ้ำ (Iterative) ที่สามารถวนเวียนและย้อนกลับเริ่มต้นใหม่ได้อีกครั้ง ทำให้แบบจำลองนี้สามารถรองรับการดำเนินงานและลดภาวะความเสี่ยงที่จะเกิดขึ้นได้เป็นอย่างดี

หัวใจของเอจายล์ (Agile Model) คือเน้นความพึงพอใจของลูกค้า โดยการส่งมอบซอฟต์แวร์ให้ลูกค้าอย่างต่อเนื่องทุก 2 สัปดาห์ ยินดีที่จะยอมรับความต้องการที่เปลี่ยนแปลงอยู่เสมอ ทีมพัฒนาระบบจะดำเนินโครงการตามขนาดของลูกค้ามีการพบหน้ากันทุกวันจนกว่าโครงการจะเสร็จ มีการประชุมสม่ำเสมอ ทีมงานมีอำนาจในการตัดสินใจเต็มที่วัดจากความก้าวหน้าของงาน การทำงานใช้กระบวนการที่ไม่หวือหวา เน้นความคุณภาพชีวิตของทีมงาน มีเทคนิคต่าง ๆ ก็นำมาแลกเปลี่ยนกัน เน้นเทคนิคการออกแบบที่ง่ายไม่ซับซ้อนทำให้บำรุงรักษาปรับเปลี่ยนระบบได้ง่าย

รูปแบบวงจรชีวิตของซอฟต์แวร์แบบเอจายด์ (Agile Software Life Cycle Mode) เป็นรูปแบบที่ออกแบบให้มีความรวดเร็ว เอจายล์พร้อมที่จะรับกับความเปลี่ยนแปลง เพื่อลดความเสี่ยงในการพัฒนาซอฟต์แวร์ โดยแบ่งเวลาออกเป็นระยะ ๆ และมีการทวนหรือรอบการพัฒนา (Iteration) แต่ละระยะซึ่งแต่ละระยะก็เปรียบเสมือนโครงการย่อย ๆ ที่มีงานภายในให้โครงการสำเร็จ ได้แก่ การวางแผนงาน การวิเคราะห์ความต้องการ

การออกแบบระบบการพัฒนาระบบ การตรวจสอบ การทำเอกสารและส่งมอบให้ลูกค้า ซึ่งแต่ละระยะจะมีงานใหม่ ๆ ออกมาและใช้เวลาไม่มากนัก ไม่เกิน 1 เดือน หรือ 4 สัปดาห์การพัฒนาจะดำเนินการอย่างต่อเนื่องถึงแม้ว่ามีอะไรมากระทบก็ไม่สนใจ เมื่อเกิดความเปลี่ยนแปลงก็จะพัฒนาให้สามารถรองรับกับความเปลี่ยนแปลงนั้นได้อย่างไม่มีข้อจำกัดตายตัว ในการพัฒนาจะเน้นการพูดคุยกันในทีมงานและผู้ใช้มากกว่าเน้นกระบวนการทำงาน (Process) หรือเครื่องมือ (Tools) การทำงานจะยึดที่ผลผลิตหรือตัวซอฟต์แวร์เป็นหลัก ไม่เน้นการจัดทำเอกสาร เน้นที่การสื่อสารเป็นหลักเพื่อให้ได้ความต้องการมาครบถ้วนและพร้อมที่จะยอมรับความเปลี่ยนแปลงเพิ่มเติมของความต้องการของผู้ใช้ ดังภาพที่ 2.8



ภาพที่ 2.8 วงจรการพัฒนาซอฟต์แวร์แบบเอจายล์

จากภาพที่ 2.8 จะเห็นว่าขั้นตอนการพัฒนาซอฟต์แวร์แบบเอจายล์ จะพยายามลดความเสี่ยงโดยพัฒนาซอฟต์แวร์ในรูปแบบของกล่องของเวลา (Time Box) สั้น ๆ เรียกว่า การทวนตั้งแต่ 1 ถึง 4 สัปดาห์ แต่ละการทวนเปรียบเสมือนโครงการซอฟต์แวร์ (Software Project) ของวงรอบเอง ซึ่งประกอบไปด้วยงานต่าง ๆ ที่จำเป็นให้เสร็จงาน ซึ่งขั้นตอนการทำงานจะเป็นไปตามวงจรการพัฒนาซอฟต์แวร์ โครงการซอฟต์แวร์แบบเอจายล์ (Agile Software Project) พยายามที่จะวนซ้ำ (Release) จะได้ซอฟต์แวร์หรือผลิตภัณฑ์ออกมาทุก ๆ การทวนและเมื่อจบแต่ละการทวนจะมีการประเมินผลด้วย การทำงานแบบเอจายล์จะนิยมการติดต่อสื่อสารแบบเวลาจริง (Real Time) คือการคุยกันต่อหน้า แทนที่จะเป็นการเขียนเอกสารโดยทีมงานจะทำงานร่วมกันกับคนเกี่ยวข้องกับการทำซอฟต์แวร์ ซึ่งอย่างน้อยจะมีนักพัฒนาระบบและลูกค้า ซึ่งได้แก่ ผู้จัดการการผลิต ผู้วิเคราะห์ธุรกิจ หรือ ผู้ใช้จริง หรืออาจจะรวมถึงผู้ทดสอบและผู้จัดการด้วย มีการเขียนเอกสารเฉพาะเท่าที่จำเป็นเพื่อให้สามารถติดต่อประสานงานได้ หลักการพื้นฐานสำคัญของระบบการทำงานแบบเอจายล์ 5 ประการ ดังนี้

ตารางที่ 2.1 หลักการพื้นฐานของการพัฒนาระบบแบบเอจายล์ 5 ประการ

หลักการพื้นฐาน	คำอธิบาย
1) ลูกค้า (Customer)	นักพัฒนาซอฟต์แวร์ ควรให้ความใกล้ชิด และติดต่อสื่อสารกับลูกค้าตลอดระยะเวลาของการพัฒนาในแต่ละขั้นตอน โดยเฉพาะขั้นตอนการจัดทำความต้องการและการประเมินผลซอฟต์แวร์ที่ได้พัฒนาขึ้น
2) ปรับปรุง (Revise)	นักพัฒนาซอฟต์แวร์ ควรมีการทบทวน แก้ไข และเพิ่มเติมคุณสมบัติของซอฟต์แวร์ที่ได้พัฒนาขึ้น จนกว่าจะบรรลุถึงความต้องการตามข้อกาหนดที่วางไว้
3) ทักษะ (Skill)	ทีมงานพัฒนาซอฟต์แวร์ ควรมีความรู้และทักษะที่เป็นประโยชน์ต่อการพัฒนา โดยปฏิบัติงานภายใต้เงื่อนไขที่วางไว้อย่างเคร่งครัด
4) เปลี่ยนแปลง (Change)	นักพัฒนาซอฟต์แวร์ควรมีความพร้อมที่จะเผชิญกับการเปลี่ยนแปลงโดยเฉพาะการออกแบบระบบที่ได้รับคำแนะนำว่าถูกต้อง เหมาะสมและสอดคล้องกับความต้องการ
5) ทำให้ง่าย (Simple)	นักพัฒนาซอฟต์แวร์ ควรมุ่งเน้นการพัฒนาซอฟต์แวร์ด้วยวิธีการอย่างง่าย โดยหลีกเลี่ยง ขั้นตอนที่ยุ่งยากและซับซ้อน

กระบวนการพัฒนาซอฟต์แวร์แบบ Agile เป็นกระบวนการพัฒนารูปแบบใหม่ มีการแบ่งรอบการพัฒนาเน้นความพึงพอใจของลูกค้าเป็นหลัก สามารถปรับเปลี่ยนได้ตามความต้องการของลูกค้า กระบวนการพัฒนาแบบ Agile มีวัตถุประสงค์หลักอยู่ 4 ข้อ (Schwaber, K., & Beedle, M., 2001) ได้แก่

1) ให้ความสำคัญกับกระบวนการมากกว่าเครื่องมือ

2) เน้นการทำงานเพื่อให้ได้ซอฟต์แวร์ที่ทำงานได้ มากกว่าการทำเอกสาร

3) เน้นการพูดคุยติดต่อสื่อสารกับลูกค้าโดยตรง มากกว่าการทำเอกสารสัญญา

4) ยอมรับการเปลี่ยนแปลงความต้องการของลูกค้า โดยไม่ยึดติดกับแผนงาน

กฎข้อกำหนดของการพัฒนาระบบแบบเอจายล์ 12 ข้อ ดังนี้

1) ความสำคัญสูงสุด คือ ความพึงพอใจของลูกค้าซึ่งสามารถเกิดขึ้นได้จากการเสนอซอฟต์แวร์ที่มีคุณค่าอย่างรวดเร็วและต่อเนื่อง ทั้งในระยะเริ่มต้นและการส่งมอบอย่างต่อเนื่องของซอฟต์แวร์ที่มีคุณค่า

2) ยอมรับการเปลี่ยนแปลงความต้องการของลูกค้า แม้ว่าการพัฒนาได้ดำเนินการไปแล้ว หรืออยู่ในส่วนท้ายของการพัฒนาเพื่อประโยชน์ของลูกค้า

3) ส่งมอบงานอย่างต่อเนื่องซึ่งควรจะมีระยะเวลาไม่น้อยกว่า 1-2 สัปดาห์ แต่ไม่มากกว่า 1-2 เดือน และปรับปรุงให้สามารถส่งงานได้ในระยะเวลาที่สั้นขึ้น

4) ทีมงานและผู้บริหารต้องมีติดต่อประสานงานทุกวัน

5) โครงการจะต้องพัฒนาด้วยทีมงานที่มีความกระตือรือร้น โดยสร้างแรงจูงใจให้แก่พนักงานจากสภาพแวดล้อมที่สนับสนุนการทำงาน และมอบสิ่งที่ทีมงานต้องการพร้อมกับเชื่อมั่นว่าทีมงานจะทำงานเสร็จ

6) การสื่อสารระหว่างทีมงานนักพัฒนา ลูกค้า และผู้บริหาร ควรจะมีการสื่อสารแบบตัวต่อตัว เพราะมีประสิทธิภาพสูง

7) ซอฟต์แวร์ที่ทำงานได้ดีเป็นเครื่องบ่งชี้ความก้าวหน้าได้ดีที่สุด

8) สนับสนุนและให้ความสำคัญกับการพัฒนาอย่างยั่งยืน โดยผู้สนับสนุนเงินทุน ผู้พัฒนา และลูกค้าได้รับความพึงพอใจร่วมกันอย่างไม่หยุดยั้ง

9) ควรปรับปรุงเทคโนโลยีที่ใช้อยู่ให้ทันสมัยอยู่เสมอและวางแผนการทำงานที่เพิ่มความเร็วในการทำงาน

10) ทำให้งานมีความเรียบง่าย การลดงานที่ไม่จำเป็น

11) สถาปัตยกรรม การรวบรวมความต้องการ และการออกแบบที่ดีที่สุดได้มาจากทีมงานที่ทำหน้าที่เฉพาะงานนั้น

12) ในระยะเวลาหนึ่งควรมีการประเมินคุณภาพการทำงานของทีมงานและปรับปรุงให้ดีขึ้นอย่างต่อเนื่อง ทีมงานสามารถสะท้อนภาพของตัวเองในเรื่องการทำงานให้ได้ผลลัพธ์ที่ดีขึ้น

เมื่อเปรียบเทียบวิธีการพัฒนาแบบ Agile กับระเบียบวิธีการพัฒนาระบบแบบดั้งเดิม (Traditional System Development Methodology) จะพบว่าวิธีการแบบ Waterfall เป็นวิธีที่ใช้กันอย่างแพร่หลาย และมีลำดับขั้นตอนที่ถูกกำหนดไว้อย่างชัดเจน โดยมุ่งหมายให้แต่ละขั้นตอนเรียงลำดับตามลำดับชั้น มีการบอกสั่งการเป็นลายลักษณ์อักษร และมีความมั่นคงในการทำงาน การทำงานในวิธีนี้เปรียบเสมือนระบบองค์กรแบบปิด (Closed System) ที่มีขั้นตอนเรียงตามลำดับ โดยเริ่มต้นจากการรวบรวมข้อมูลและกำหนดความต้องการของผู้ใช้จนถึงขั้นตอนการออกแบบ การเขียนโปรแกรม การทดสอบ และการติดตั้งระบบ

แต่การพัฒนาแบบเอจายล์การทำงานจะสั้น ยึดเป้าหมายงานมากกว่าระเบียบ การสื่อสารทำได้รวดเร็วและเป็นองค์กรระบบเปิด (Opened System) จะแบ่งงานออกเป็นส่วนย่อย ๆ แล้วค่อย ๆ ทดสอบไปเรื่อย ๆ ทุกสัปดาห์หรือทุกสองสัปดาห์ ทั้งนี้จะเน้นสร้างส่วนย่อยที่สุดของงานทั้งหมดที่สามารถใช้งานได้ทีละชิ้นเพื่อให้ส่งมอบได้รวดเร็ว และทำการปรับปรุงเพิ่มเติมอย่างต่อเนื่องตลอดระยะเวลาของโครงงาน สามารถเปรียบเทียบวิธีการพัฒนาแบบเอจายล์กับวิธีการพัฒนาแบบน้ำตกได้ สามารถเปรียบเทียบประเด็นต่าง ๆ ระหว่างการบริหารจัดการโครงการแบบดั้งเดิมและการบริหารจัดการแบบเอจายล์

ตารางที่ 2.2 การเปรียบเทียบระหว่างการจัดการโครงการแบบดั้งเดิมและแบบเอจายล์

คุณลักษณะ	**การบริหารจัดการแบบดั้งเดิม (แบบจำลองน้ำตก)**	**การบริหารจัดการแบบเอจายล์**
ลักษณะโครงการ	มั่นคง (Stable)	เป็นพลวัต (Dynamic)
การปรับเปลี่ยน	ไม่ยืดหยุ่น (Inflexible)	ยืดหยุ่น (Flexible)
การให้ความสำคัญ	ให้ความสำคัญกับงาน (Job-focused)	ให้ความสำคัญกับทักษะ (Skills-focused)
ตำแหน่งการทำงาน	งานจะต้องมีการระบุตำแหน่ง	งานจะต้องระบุถึงสิ่งที่จะต้องทำ
ความสำคัญทีมงาน	เน้นคน (Individual – oriented)	เน้นทีมงาน (Team – oriented)
ลักษณะงาน	งานถาวร (Permanent jobs)	งานชั่วคราว (Temporary job)
การทำงาน	เน้นการบังคับบัญชา	เน้นการมีส่วนร่วม
การตัดสินใจ	ผู้จัดการเป็นผู้ตัดสินใจ	พนักงานมีส่วนร่วมในการตัดสินใจ
การบริหารจัดการ	การบริหารจัดการงานตามลำดับ ขั้นตอนตายตัว	การบริหารจัดการงานง่าย ยืดหยุ่น
ระยะเวลา	ใช้ระยะเวลามาก	ใช้ระยะเวลาน้อย
เอกสาร	เน้นการทำงานมากกว่าเอกสาร	เน้นการทำงานเป็นเอกสาร

สรุปได้ว่าวิธีการพัฒนาแบบเอจายล์เป็นหลักการในการพัฒนาระบบแบบใหม่ ที่เน้นให้การทำงานรวดเร็ว และมีการเตรียมพร้อมที่จะตอบสนองต่อความเปลี่ยนแปลงที่อาจเกิดขึ้นได้ตลอดเวลา โดยเฉพาะความต้องการของผู้ใช้ วิธีพัฒนาแบบนี้ถือเป็นการพัฒนาแบบทำซ้ำ ที่จะต้องมีการพบปะสนทนากับผู้ใช้อยู่ตลอดเวลา และในขณะที่พบปะกันนั้นก็จะถือเป็นระยะเวลาของการส่งงานไปในตัวด้วย การส่งงานแต่ละครั้งของวิธีการพัฒนาแบบนี้ โดยส่วนใหญ่จะเป็นการพัฒนาเฉพาะส่วนย่อย ๆ แล้วค่อยทยอยส่งให้กับผู้ใช้ เมื่อผู้ใช้ได้ทดสอบหรือประเมินระบบแล้ว ถ้าต้องการปรับเปลี่ยนตรงส่วนใดก็สามารถทำได้โดยที่ไม่ต้องรื้อระบบใหม่ทั้งหมด

1. วิธีการพัฒนาซอฟต์แวร์แบบเอจายล์

ปัจจุบันมีกระบวนการพัฒนาแบบ Agile Model จำนวนมาก เช่น Extreme Programming (XP), Scrum, FDD (Feature Driven Development), DSDM (Dynamic System Development Method) แต่ที่นิยมนำมาใช้ในกระบวนการพัฒนาซอฟต์แวร์ขนาดเล็กถึงขนาดกลางจะเป็น Extreme Programming และ Scrum (Beck, K. & Andres, C., 2004) สามารถอธิบายวิธีการพัฒนาแต่รูปแบ่งได้ดังต่อไปนี้

1) วิธีการพัฒนาแบบ Extreme Programming (XP)

Beck, K. & Andres, C. (2004). XP เป็นแบบจำลองที่ได้รับความนิยมในการนำไปใช้ในกระบวนการพัฒนาซอฟต์แวร์ ซึ่งมีหลักการปฏิบัติในภาพรวม เช่น โปรแกรมเมอร์ มีจำนวน 5 - 10 คน ทำงานในสถานที่เดียวกันกับลูกค้า และพัฒนาแบบ Pair programming การพัฒนาจะแบ่งเป็นรอบ ส่งงานเป็น release และมีการเพิ่มฟังก์ชั่นการพัฒนาในทุกรอบ ๆ และ Requirement จะถูกเขียนใน User stories เป็นต้น

XP ตัวอักษร "P" มาจาก Programming เน้นหลักปฏิบัติไปที่การพัฒนา วิธี XP เหมาะกับการพัฒนาซอฟต์แวร์ที่มีผู้พัฒนาไม่เกิน 10 คน สามารถพูดคุยสอบถามได้อย่างรวดเร็ว

XP เป็นแบบจำลองหนึ่งของกระบวนการพัฒนาแบบ Agile ที่มีผู้ยอมรับและนำมาใช้มาก มีหลักปฏิบัติอยู่ 12 ข้อ (Svensson, H., & Höst, M.,2005) ได้แก่

1.1) Pair Programming ผู้พัฒนาจับคู่กันพัฒนาซอฟต์แวร์โดยใช้หน้าจอเดียวกัน คนหนึ่งเขียนโปรแกรม อีกคนช่วยดูหรือสามารถสลับบทบาทกันได้

1.2) Common Code Ownership ซอฟต์แวร์ที่เขียนขึ้นมาทุกบรรทัด ให้ถือว่าเป็นของส่วนรวม ทุกคนมีสิทธิ์เข้าไปดูและแก้ไขได้เสมือนเป็นของตัวเอง

1.3) Planning Game มีการวางแผนการทำงานเสมือนว่าทุกคนกำลังจะเล่นเกม ทุกคนอยู่ในทีมเดียวกัน มีบทบาทและหน้าที่แตกต่างกัน ตามที่ได้รับมอบหมายหรือตกลงกัน เพื่อบรรลุเป้าหมายคือชนะเกม หรือพัฒนาซอฟต์แวร์สำเร็จ

1.4) Frequent Small Releases มีการแบ่งรอบการพัฒนา ทำให้ซอฟต์แวร์ถูกแบ่งออกเป็นส่วน ๆ ทำให้สามารถส่งมอบให้ลูกค้าเห็นได้บ่อยขึ้น

1.5) Continuous Integration เนื่องจากการพัฒนาถูกแบ่งออกเป็นรอบส่งผลให้ซอฟต์แวร์ที่ได้จะเป็นชิ้นส่วน จึงจำเป็นต้องมีการทำการรวมซอฟต์แวร์ให้เป็นชิ้นเดียวอย่างต่อเนื่องสม่ำเสมอ และทุกครั้งที่มีการผลิตชิ้นส่วนต้องมีการรวมซอฟต์แวร์ให้เป็นชิ้นเดียวกัน

1.6) Refactoring Code เมื่อมีการพัฒนาซอฟต์แวร์ได้ระยะหนึ่ง โค้ดของการพัฒนาส่วนใหญ่จะไม่เป็นระเบียบ ทำให้ต้องมีการปรับแต่งเมื่อมีโอกาส เช่น การ Extract interface จาก Class ต่างๆ

1.7) On-site Customer เนื่องจาก XP จะเน้นความพึงพอใจของลูกค้า ดังนั้นจึงให้ทีมนักพัฒนาไปอยู่ที่ Site เดียวกับลูกค้า เพื่อที่จะสามารถติดต่อสื่อสารกับลูกค้าได้มากขึ้น และสามารถรับรู้ปัญหาเพื่อเข้าถึงความต้องการของลูกค้าได้จริง

1.8) Coding Standard การพัฒนาด้วยกระบวนการแบบ XP จำเป็นต้องมีการกำหนดมาตรฐานของการพัฒนาโค้ดให้ตรงกัน เนื่องจากมีผู้พัฒนาหลายคน เพื่อให้สอดคล้องกับหลักปฏิบัติเรื่อง common code ownership และผู้ที่จะนำไปพัฒนาหรือปรับปรุงต่อสามารถเข้าใจซอฟต์แวร์ได้ง่ายขึ้น

1.9) Sustainable Pace วิธี XP จะไม่เน้นให้ผู้พัฒนาทำงานหักโหมเกินไป ให้พัฒนาอย่างต่อเนื่องแบบสม่ำเสมอระบุไว้ว่าไม่ควรเกิน 40 ชั่วโมงต่อสัปดาห์

1.10) Test Driven Development มีการเขียน Test Case ขึ้นมาก่อนพัฒนาซอฟต์แวร์เมื่อการพัฒนาสิ้นสุดก็จะทดสอบกรณีทดสอบ เพื่อดูว่าผลลัพธ์ของซอฟต์แวร์ที่ได้ถูกต้องกรณีทดสอบหรือไม่ เพื่อลดความผิดพลาดของการพัฒนา

1.11) Simple Design เนื่องจากการพัฒนาด้วย XP จะเน้นที่ลูกค้าสามารถเปลี่ยนแปลงซอฟต์แวร์ได้ตามต้องการในขณะที่พัฒนาอยู่ ทำให้การออกแบบต้องมีความยืดหยุ่นง่ายต่อการปรับเปลี่ยน จึงให้การออกแบบในระยะเริ่มต้นไม่ต้องใส่เทคนิคหรือลูกเล่นมาก

1.12) System Metaphor เป็นการเปรียบเทียบให้เห็นการทำงานของซอฟต์แวร์ที่จะพัฒนา เช่น นำไปเปรียบเทียบกับซอฟต์แวร์ที่เคยมีประสบการณ์ในการพัฒนามาก่อน เพื่อให้ผู้พัฒนามองเห็นภาพ และพัฒนาได้ง่ายขึ้น

2) วิธีการพัฒนาแบบ Scrum

Scrum (Schwaber, K., & Beedle, M., 2001) ถูกพัฒนาขึ้นเมื่อปี 1995 เป็นกระบวนการพัฒนาแบบ Agile เป็นวิธีการที่ค่อนข้างเป็นที่นิยมและถูกนำไปใช้มากที่สุดวิธีการหนึ่ง ซึ่งจะกำหนดรอบของการพัฒนา (Sprint) คล้ายกับ XP แต่จะกำหนดจำนวนวันที่ดำเนินการในแต่ละรอบ (Sprint) ไว้สูงสุดที่ 30 วันเสมอ และจะ

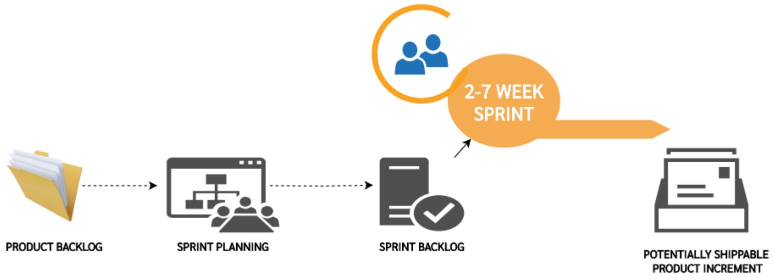
เลือกฟังก์ชั่นการทำงานเข้าไปในรอบ (Sprint) ที่เหมาะสมกับจำนวนวันดังกล่าว โดยหลังจากที่วิเคราะห์ความต้องการออกมาเป็นฟังก์ชั่นการทำงานของระบบแล้ว จะถูกเรียก Backlogs เตรียมที่จะนำเข้าไปสู่รอบการพัฒนา (Sprint)

SCRUM เป็นวิธีการการพัฒนาซอฟต์แวร์ของ Agile หนึ่ง ซึ่งในกระบวนการของวิธีนี้ ค่อนข้างเป็นที่นิยมและถูกนำไปใช้มากที่สุดตัวหนึ่ง โดย SCRUM จัดเป็นวิธีที่มีขั้นตอนการทำงานที่ชัดเจนพอสมควรและเป็นกระบวนการที่ไม่เข้มงวดจนเกินไป ถ้าใครที่มีประสบการณ์ในการพัฒนาซอฟต์แวร์โดยใช้ Waterfall Model มาก่อน จะทำความเข้าใจกับ SCRUM ได้ง่ายกว่าเมื่อเทียบกับวิธีของ XP หลักการพื้นฐานของ SCRUM มาจากวิธีการ Iteration และ Incremental เช่นเดียวกับกระบวนการ ทุกตัวของ Agile ที่เน้นให้การพัฒนาทำเป็นรอบสั้นๆ เพื่อให้สามารถส่งมอบซอฟต์แวร์ได้เร็ว ตามลำดับความสำคัญที่ลูกค้ากำหนดมาให้ ซึ่ง SCRUM จะมีจุดเด่นในเรื่องของกระบวนการและระยะเวลาในการทำงานหรือ time-box ที่ชัดเจน รวมทั้งมีหน้าที่ต่างๆ ที่ชัดเจน Scrum Roles วิธีการของ SCRUM จะแบ่งหน้าที่ต่างๆ ออกเป็น 3 ส่วน ได้แก่

2.1) Product Owner คือลูกค้า เนื่องจาก SCRUM จะไม่เรียกว่า Customer เพราะต้องการแสดงความเป็นเจ้าของเป็นผู้สนับสนุนทุนในการพัฒนาซอฟต์แวร์ และเป็นผู้กำหนดว่าซอฟต์แวร์ที่ต้องการนั้นจะเป็นอย่างไร อะไรคือสิ่งที่มีคุณค่ต่อธุรกิจ ดังนั้นถ้าลูกค้าไม่มีส่วนร่วมในการทำงาน ไม่เข้าใจและไม่ให้ความสำคัญการพัฒนาก็จะมีปัญหา

2.2) The Team หรือทีมพัฒนา ซึ่งทีมในความหมายของ SCRUM ควรจะมีทุกตำแหน่งที่จำเป็นและครบถ้วน เช่น SA, Programmer, Tester และอื่น ๆ โดยไม่มีการแยกออกไปเป็นอีกทีมหนึ่ง ทุกคน คือทีมเดียวกัน สามารถออกความเห็นได้ทุกเรื่อง และทั้งทีมมีหน้าที่ที่จะต้องช่วยกันพัฒนา software ให้สำเร็จให้ได้ อะไรที่เป็นปัญหาก็ช่วยกันทำให้เสร็จ (scrum)

2.3) Scrum Master มีหน้าที่ดูแลให้ 2 ส่วนแรกทำงานร่วมกันได้ เพื่อให้การพัฒนาซอฟต์แวร์สำเร็จลุล่วงไปได้ตามที่ลูกค้าต้องการ และมีหน้าที่คอยแนะนำกฎต่าง ๆ ของ SCRUM ให้แก่ ทีมพัฒนาและลูกค้ารวมทั้งคอยอำนวยความสะดวกให้ด้วย จะคล้ายผู้จัดการโครงการ แต่ไม่เหมือนกัน เพราะในทีม SCRUM นั้นจะไม่มีหัวหน้า สมาชิกทุกคนในทีมจะต้องมีความรับผิดชอบ ทุกคนเป็นหัวหน้าของตัวเอง และเป็นหัวหน้าของคนอื่น ๆ ในทีมด้วย เมื่อไม่มีคนคอยบอกว่าใครต้องทำอะไร สมาชิกในทีมก็จะต้องกระตือรือร้นมากกว่าเดิม ซึ่งถ้าในทีมพัฒนาไม่มีวุฒิภาวะมากพอ Agile จะสำเร็จได้ยาก วิธีการแบบ SCRUM ลูกค้าอาจไม่ต้องอยู่กับเราตลอดเวลาแบบวิธีการของ XP แต่จะมีการประชุมและคอยตรวจสอบสถานะของโครงการบ่อย ๆ โดยกำหนดวันเวลาที่จะตรวจสอบที่แน่นอน แต่เมื่อเกิดปัญหาหรือมีคำถาม ก็ยังจำเป็นที่ลูกค้าจะต้องให้ทีมพัฒนาสามารถติดต่อได้ตลอดเวลาเช่นกัน เพื่อเป็นผู้ตัดสินและกำหนดทางเลือกที่ถูกต้องให้



ภาพที่ 2.9 ขั้นตอนการทำงานของ SCRUM

จากภาพ 2.9 ส่วนที่เป็นวนซ้ำ Iteration ก็คือลักษณะการทำงานโดยแบ่งระยะเวลาการทำงานทั้งหมดออกเป็นรอบ ๆ มีการกำหนดเวลาที่ชัดเจนและเท่า ๆ กันทุกรอบ (ในรูปคือ 2-4 weeks) ส่วน Incremental ก็เป็นการแบ่งความต้องการทั้งหมดออกเป็นส่วนย่อย ๆ แล้วค่อย ๆ ทำไปทีละความต้องการย่อย ๆ ที่เขียนว่า Product Backlog สามารถอธิบายขั้นตอนการทำงานของ SCRUM มีดังต่อไปนี้

1) พัฒนา Product Backlog ขึ้นมาก่อน (ความต้องการทั้งหมด) โดยให้ทางลูกค้า ซึ่งก็คือ Product Owner เขียนขึ้นมา และจะต้องกำหนดลำดับความสำคัญให้กับแต่ละ Product Backlog ซึ่งสิ่งที่ได้รับคือข้อกำหนดแผนงานของทั้งโครงการ

2) วางแผนการพัฒนาทั้งโครงการคร่าว ๆ เรียกว่า Release Planning Meeting เพื่อประมาณการว่าระยะเวลาของแต่ละรอบ iteration จะกำหนดที่กี่ man/day ตัวโครงการนี้จะใช้เวลากี่ iteration ในแต่ละ iteration จะได้อะไรบ้าง การประเมินจะเกิดขึ้นในช่วงนี้ เทคนิคที่ใช้ในการประเมินแตกต่างจากวิธีคิดแบบ man/day (ideal day) เรียกว่า Story Point (สำหรับในกระบวนการในข้อ 1-2 อาจจะเรียกว่า iteration 0 (ศูนย์) หรือใน XP จะเรียกว่า Architectural Spike ซึ่งในข้อ 1-2 ปรกติจะไม่จบในครั้งเดียว เพราะอาจจะมีอะไรที่ไม่ครอบคลุมกล่าวเอาไว้ ในระหว่างทำจึงค่อยมาปรับแก้ได้ ดังนั้นจะไม่ใช้เวลานาน)

3) ทำการวางแผนงานในแต่ละรอบ Sprint (iteration) หรือจะเรียกว่า Sprint Planning Meeting โดยการพัฒนาเอกสาร เรียกว่า Sprint Backlog ซึ่ง Sprint Backlog จะเป็นงานต่าง ๆ ที่ทางทีมจะต้องสร้าง เพื่อให้ Product Backlog แต่ละข้อนั้นกลายเป็นซอฟต์แวร์ที่ใช้งานได้จริง

4) เมื่อกำหนดเรียบร้อย จะเป็นขั้นตอนการยืนยันกับลูกค้าอีกครั้ง เพื่ออธิบายแผนและภาพรวมวิธีการต่าง ๆ และแจ้งให้ลูกค้าทราบว่าเมื่อจบ iteration แรก ลูกค้าจะได้อะไรบ้างและจะเป็นการยืนยันแบบนี้ไปทีละ iteration กรณีที่ลูกค้ายืนยันแล้ว Product Backlog จะต้องถูกกำหนดให้ไม่มีการเพิ่มหรือลดคุณสมบัติหลักเพื่อจะได้เข้าสู่ขั้นตอนพัฒนา

5) ในการพัฒนาในแต่ละวันจะมีการประชุมสั้น ๆ ไม่เกิน 15 นาที เพื่อรายงานสถานะการดำเนินงาน เรียกว่ Daily Scrum Meeting (หรือ Standup meeting) โดยมีส่วนเครื่องมือที่ใช้ติดตามความก้าวหน้าของงาน คือ Bur-Down Chart, Burr-Up Chart, Task Board

6) เมื่อถึงกำหนดส่งงานก็จะมีการประชุมกันระหว่างทีมพัฒนาและลูกค้าเพื่อตรวจสอบ สิ่งที่ได้จาก iteration นั้น ไม่เกิน 4 ชั่วโมง เรียกว่า Sprint Review Meeting ซึ่งลูกค้าจะต้องตรวจสอบว่าเป็นไปตามความต้องการหรือไม่ ซึ่งระบบจะต้องแสดงคุณสมบัติตามที่ตกลงกันไว้ในข้อ 4 ได้ในระดับที่ใกล้เคียงกับ production มากที่สุด

7) เมื่อตรวจสอบจบแล้ว ทีมพัฒนาก็จะกลับมาประชุมกันเองอีกครั้ง เพื่อนำข้อเสนอแนะหรือความเห็นจากลูกค้มาสรุป รวมทั้งนำสิ่งที่ผ่านมาใน iteration ที่แล้วมาคุยกัน ว่ามีอะไรที่เป็นปัญหาควรจะต้องนำมาปรับปรุงแก้ไขบ้าง หรือที่เรียกว่า Retrospective Meeting ใช้เวลาไม่เกิน 3 ชั่วโมง

8) เมื่อได้ข้อสรุปทั้งหมด ก็นำข้อมูลเหล่านี้ไปปรับแผนใน iteration ถัดไป โดยวนกลับไปที่ข้อ 3 อีกครั้ง สิ่งที่ยังทำไม่เสร็จใน iteration ที่แล้ว เช่น bug หรือคุณสมบัติอื่น ๆ ที่ทำไม่ทัน จะต้องถูกนำมาทำก่อนเรื่องอื่นใน iteration นี้

9) ทำซ้ำไปเรื่อย ๆ จนจบโครงการ

ทั้งวิธีการพัฒนารแบบ Scrum และ XP ต่างเป็นกระบวนการที่อยู่บนพื้นฐานของ Agile มีคุณลักษณะ (characteristics) ที่เหมือนกัน คือ มีช่วงระยะเวลาการพัฒนาในแต่ละรอบที่สั้น (Iteration/Sprint) และมีหลายรอบ ซึ่งมีผู้มักเข้าใจผิดว่า Scrum และ XP คือตัวเดียวกัน ซึ่งความจริงแล้วไม่ใช่ Scrum จะเน้นหลักปฏิบัติไปที่การบริหารการพัฒนาซอฟต์แวร์ (Management Methodology) มากกว่าการอธิบายเทคนิคการพัฒนาซอฟต์แวร์ (Engineering Methodology) หลักปฏิบัติของ Scrum เป็นเพียงส่วนหนึ่ง (Subset) ของ XP (Schwaber, Ken, 2004) XP มีหลักปฏิบัติที่เสริมด้านเทคนิคการพัฒนาซอฟต์แวร์เข้าไปด้วย (Engineering Methodology) เช่น Pair programming, Test-driven development, Refactoring เป็นต้น (Beck, K. & Andres, C., 2004)

3) วิธีการพัฒนาซอฟต์แวร์แบบปรับตัว (Adaptive Software Development: ASD)

วิธีการพัฒนาซอฟต์แวร์แบบปรับตัว (เอเอสดี ถูกนำเสนอโดย Jim Highsmith สร้างขึ้นเพื่อสนับสนุนการพัฒนาระบบที่ซับซ้อน ให้ความสำคัญกับการทำงานร่วมกันระหว่างบุคคล และการจัดระเบียบทีมงาน โดยลักษณะการทำงานแบ่งออกเป็น 3 ส่วนหลัก ดังนี้

3.1) การคาดเดา (Speculation) เริ่มต้นการวางแผนโดยใช้ข้อมูลเบื้องต้น ได้แก่ ข้อมูลเกี่ยวกับภารกิจของลูกค้า เงื่อนไขต่าง ๆ ของโครงการและความต้องการพื้นฐาน เพื่อกำหนดชุดของวงจรการวงรอบสำหรับซอฟต์แวร์ที่โครงการจะต้องผลิต

3.1.1) การร่วมมือ (Collaboration) การร่วมมือกันระหว่างผู้ที่เกี่ยวข้องเป็นวิธีที่ใช้ในทุก ๆ วิธีการแบบ Agile สำหรับการพัฒนาซอฟต์แวร์แบบปรับตัว (เอเอสดี) ให้ความสำคัญกับการทำงานร่วมกันดังนี้

1) การวิพากษ์วิจารณ์ชิ้นงาน

2) การช่วยเหลือกัน

3) การกระจายงานอย่างเท่าเทียมกัน

4) สร้างความชำนาญเฉพาะที่จะเป็นประโยชน์ต่องาน

5) มีการพูดคุยถึงปัญหาหรือข้อกังวลสงสัยเพื่อนำไปสู่การทำงานที่ได้ประสิทธิผล

3.1.2) การเรียนรู้ (Learning) High smith อ้างว่าทีมพัฒนาซอฟต์แวร์มักจะประมาณความเข้าใจของตนเองด้านเทคโนโลยีและด้านอื่น ๆ มากเกินจริง ดังนั้นสมาชิกของทีมเอเอสดีเริ่มพัฒนาชิ้นส่วนที่เป็นส่วนหนึ่งของวงจรปรับตัว ทีมพัฒนาซอฟต์แวร์ควรเรียนรู้ไปพร้อม ๆ กับความก้าวหน้าของกระบวนการพัฒนา โดยกระบวนการเรียนรู้มี 3 วิธีคือ

1) กลุ่มเฉพาะทาง (Focus Groups) เมื่อลูกค้า/ผู้ใช้งานให้ผลตอบกลับในการใช้งาน ในแต่ละรุ่นของซอฟต์แวร์ที่ส่งมอบไป สิ่งนี้จะเป็นตัวบ่งชี้ว่าผลิตภัณฑ์ได้ตอบสนองความจำเป็นทางธุรกิจหรือไม่

2) การทบทวนทางเทคนิคอย่างเป็นทางการ (Formal Technical Review) ทีมเอเอสดีมีการทบทวนชิ้นส่วนซอฟต์แวร์ที่กำลังพัฒนาอยู่ ขณะเดียวกันมีการปรับปรุงคุณภาพและเรียนรู้ไปพร้อม ๆ กัน

3) การตรวจสอบภายหลัง (Postmortems) เมื่อเสร็จสิ้นงานแล้ว ทีมงานเอเอสดีกลายเป็นผู้วิจารณ์งานและคุณภาพของตนเอง ซึ่งเป็นเป้าหมายของการเรียนรู้เพื่อปรับปรุงวิธีการทำงาน

4) วิธีการพัฒนาระบบแบบไม่หยุดนิ่ง (Dynamic System Development Method: DSDM)

วิธีการพัฒนาระบบแบบไม่หยุดนิ่ง (ดีเอสดีเอ็ม) เป็นการพัฒนาซอฟต์แวร์ Agile ที่มีการกำหนดกรอบงานสำหรับการพัฒนาโดยมีข้อจำกัดด้านเวลา ซึ่งใช้การพัฒนาต้นแบบและพัฒนาเพิ่มขึ้น ในสิ่งแวดล้อมโครงการที่มีการควบคุม โดยพยายามพัฒนาขึ้นงาน 80 เปอร์เซ็นต์ของระบบให้เสร็จภายในเวลา 20 เปอร์เซ็นต์ ของเวลาทั้งหมดที่ใช้ในการพัฒนา

ดีเอสดีเอ็มเป็นกระบวนการวงรอบ (Iteration) เช่นเดียวกับ XP และ ASD มีความแตกต่างตรงที่แต่ละรอบของวงจรดีเอสดีเอ็ม จะเป็นไปตามกฎ 80 เปอร์เซ็นต์ คือ ทำงานให้เท่าที่จำเป็นในแต่ละวงรอบ เพื่อให้เคลื่อน

ไปสู่วงรอบถัดไป ส่วนรายละเอียดที่เหลือสามารถทำให้เสร็จได้ภายหลัง เมื่อทราบความต้องการทางธุรกิจเพิ่มเติม หรือเมื่อได้รับการร้องขอให้เปลี่ยนแปลงโดยลักษณะการทำงานแบ่งออกเป็น 5 ส่วนหลัก ดังนี้

4.1) การศึกษาความเป็นไปได้ (Feasibility Study) สร้างรายการความต้องการและข้อกำหนดทางธุรกิจพื้นฐานของแอพพลิเคชั่นที่จะสร้าง ซึ่งมีการประเมินว่าระบบมีความเป็นไปได้ สำหรับกระบวนการดีเอสดีเอ็มหรือไม่

4.2) การศึกษาด้านธุรกิจ (Business Study) จัดสร้างความต้องการเชิงข่าวสาร และเชิงหน้าที่ของระบบ นิยามสถาปัตยกรรมและระบุความต้องการในการบำรุงรักษาระบบ

4.3) การทำวนซ้ำแบบจำลองเชิงหน้าที่ (Function Model Iteration) พัฒนาต้นแบบและพัฒนาเพิ่มขึ้น และวิวัฒนาการไปเป็นระบบที่ส่งมอบได้ การทำวนซ้ำเป็นการรวบรวมความต้องการเพิ่มเติมจากการตอบสนองกลับมาของผู้ใช้

4.4) การทำวนซ้ำการออกแบบและการสร้าง (Design and Build Iteration) ต้นแบบที่พัฒนาขึ้นระหว่างกระบวนการวนซ้ำการสร้างแบบจำลองเชิงหน้าที่ เพื่อให้มั่นใจว่าซอฟต์แวร์สามารถทำงานได้จริงบางกรณี การทำวนซ้ำการสร้างแบบจำลองเชิงหน้าที่และการทำวนซ้ำการออกแบบและการสร้างเกิดขึ้นพร้อม ๆ กัน

4.5) การอิมพลีเมนต์ (Implementation) การใช้งานซอฟต์แวร์ภายใต้สิ่งแวดล้อมการทำงานจริง 1) ซอฟต์แวร์อาจไม่ต้องสมบูรณ์ร้อยเปอร์เซ็นต์ หรือ 2) การร้องขอการเปลี่ยนแปลงอาจเกิดขึ้นได้ขณะใช้งานอยู่ในกรณีใดกรณีหนึ่งการพัฒนาดีเอสดีเอ็มจะดำเนินต่อเนื่องไป โดยย้อนกลับมาทำกิจกรรมการวนซ้ำ

5) วิธีการพัฒนาแบบคริสตัล (Crystal)

Alistair Cockbum และ Jim Highsmith ได้สร้างแบบจำลองคลิสตัล (Crytal) ของวิธีการเอจายล์ (Agile) เพื่อให้ได้มาซึ่งวิธีการพัฒนาซอฟต์แวร์ที่เน้นความสามารถในการพัฒนาแบบนำร่อง ได้แก่ โครงการที่มีทรัพยากรจำกัด ทีมงานต้องร่วมมือกันในการสื่อสาร โดยมีเป้าหมายหลัก คือ ส่งมอบซอฟต์แวร์ที่มีประโยชน์ทำงานได้ และเป้าหมายรอง คือ จัดเตรียมความพร้อมสำหรับการวงรอบถัดไป

6) วิธีการพัฒนาที่ขับเคลื่อนด้วยคุณลักษณะของซอฟต์แวร์ (Feature Driven Development: FDD)

วิธีการพัฒนาที่ขับเคลื่อนด้วยคุณลักษณะของซอฟต์แวร์ (เอฟดีดี) เริ่มขึ้นจากแนวคิดของ Peter Coad และคณะ เพื่อสร้างแบบจำลองกระบวนการเชิงปฏิบัติการของวิศวกรรมซอฟต์แวร์เชิงวัตถุ จากนั้น Stephen Palmer และ John Felsing ได้พัฒนาเพิ่มเติมงานของ Coad เพื่อประยุกต์กับโครงการขนาดขนาดใหญ่ขึ้น

เนื่องจากคุณลักษณะเป็นส่วนเล็ก ๆ ของซอฟต์แวร์ที่ทำงานได้ จึงสามารถอธิบายโครงสร้างและความสัมพันธ์ระหว่างกันได้ง่ายกว่า อีกทั้งยังสามารถทบทวนได้ดีกว่าหากเกิดข้อผิดพลาดจึงง่ายต่อการตรวจทานอย่างละเอียด เอฟดีดีให้รายละเอียดและเทคนิคของการจัดการโครงการมากกว่าวิธี Agile แบบอื่น ๆ เมื่อโครงการมีความซับซ้อนมากขึ้น การจัดการโครงการเฉพาะกิจอาจไม่เพียงพอ จึงจำเป็นที่ผู้ที่เกี่ยวข้องต้องเข้าใจสถานะของโครงการว่าได้บรรลุเป้าหมายอะไรไปแล้วบ้าง และมีปัญหาอะไรบ้างที่พบ กิจกรรมการพัฒนาระบบแบ่งเป็น "การออกแบบคร่าว ๆ การออกแบบ การตรวจทานการออกแบบ การโค้ด การตรวจทานการโค้ด และการส่งเสริมการสร้าง" การเปรียบทียบแนวปฏิบัติหลักกับแถลงการณ์ของเอจายล์ (Agie)

2. ปัญหาและอุปสรรคของวิธีการพัฒนาระบบแบบเอจายล์

ส่วนที่ยากที่สุดในการดำเนินการโครงการประเภท Iteration และ Incremental ก็คือการวางแผนและประมาณการเพราะมักจะเป็นสิ่งที่ลูกค้าถามอยู่เสมอ (Softwire Development Estimating & Planning) ในรายละเอียดต่อไปนี้ จะเป็นการแสดงวิธีการการประเมินโดยทั่วไปจนถึงวิธีการที่เหมาะสมสำหรับวิธี Agile (Case XP Methodology)

1) การประเมินจากงานทั้งหมด

ที่ผ่านมาการวางแผนและประมาณการ จะทำโดยการแตกสิ่งที่ทีมพัฒนาต้องทำตามความต้องการออกมาเป็นชิ้นงาน (Task) แล้วประเมินว่าแต่ละชิ้นงานนั้นใช้เวลาเท่าไหร่ และแจกงานให้แต่ละคนเอา ชิ้นงานเหล่านั้นไปทำงาน กรณีที่ขึ้นงานใดที่สามารถทำพร้อมกันได้ ก็เอามาวางแบบคู่ขนานชิ้นงานใดที่ต้องรอชิ้นงานอื่น ต้องนำไปต่อท้ายชิ้นงานนั้น สุดท้ายก็ได้เป็น Gantt chart ที่บอกว่าจากขึ้นงานทั้งหมดงานจะแล้วเสร็จเมื่อใด

2) การประเมินพิจารณาจากความสามารถ

ปัญหาของการวางแผนแบบนี้ ข้อแรก ก็คือ เราประเมินงานด้วยระยะเวลาอย่างเดียว แต่ความสำคัญอยู่ที่จำนวนวันที่คนคนหนึ่งใช้ในการพัฒนา จะไม่เท่ากับจำนวนวันที่อีกคนหนึ่งใช้ในการพัฒนา หรืองานแบบเดียวกันใช้วิธีการเขียนโปรแกรมเหมือน ๆ กันไม่ได้ เพราะแต่ละคนก็จะมีความถนัดและประสบการณ์ที่ต่างกัน

3) ประเมินจากความสำคัญของธุรกิจ

ลูกค้าจะได้ประโยชน์อะไรจากการวางแผนตามชิ้นงานเหล่านั้น เพราะลูกค้าให้ความต้องการแต่ละข้อใน Product backlog ซึ่งโดยปกติแล้วลูกค้าจะไม่สนใจว่าความต้องการแต่ละข้อนั้นจะถูกแตกงานออกไปกี่ชิ้นงานและใครเป็นคนทำชิ้นงานนั้น ๆ สิ่งที่ลูกค้าจะสนใจคือการติดตามผลงานตามความสำคัญของความต้องการเหล่านั้น

จากข้อ 1-3 ทำให้แนวทางในการวางแผนและประมาณการของ Agile ก็ยังคงอยู่บนพื้นฐานของ Iterative & Incremental เช่นเดิม แต่การแตกช่วงของการพัฒนาเป็น iteration นั้นเราจะต้องมีวิธีการวางแผนและประมาณการที่สอดคล้องกับลักษณะของ iteration ด้วย ซึ่ง Story Point จะเป็นเครื่องมือพื้นฐานชิ้นแรกที่จะทำให้การวางแผนและติดตามงานแบบ iteration & Incremental ให้เป็นไปได้ในทางปฏิบัติ

4) การให้คะแนนในแต่ละความต้องการ (Story Point)

การกำหนดขนาดความต้องการระบบ ซึ่งถ้าความสามารถของแต่ละคนแตกต่างออกไป แต่ขนาดของความต้องการก็ยังคงเท่าเดิมเสมอ ประเมินขนาดจะนำมาใช้ในการประเมินควบคู่ไปกับการประเมินจำนวนวันที่จะใช้ในการทำงานการประเมินคะแนน (Point) ให้กับแต่ละความต้องการแต่ละข้อนั้นไม่มีกฎเกณฑ์ตายตัว แต่จะให้ทุกคนช่วยกันประเมินร่วมกัน และได้หลักในการประเมินข้อเดียว แล้วใช้หลักนั้นในประเมินข้อแรกนั้นนำไปใช้ในลำดับถัด ๆ ไป

5) นำขนาดความต้องการมาประเมินความสามารถของทีม (Velocity)

Velocity จะเป็นค่าที่เอาไว้ประเมินความสามารถของทีม โดยไม่ประเมินรายบุคคล เพื่อใช้วัดว่าทีมของเรานั้นจะสามารถพัฒนาซอฟต์แวร์ได้กี่ Story Point ต่อ Iteration การนำ Story Point มาประเมินเวลานั้นอาจจะทำได้ยากและใช้เวลามากกว่าในช่วงแรก ๆ เพราะการขาดประสบการณ์ในการประมาณขนาดของความต้องการ แต่เมื่อเริ่มทำไปจะพบว่าการประเมินด้วยคะแนน จะทำง่ายกว่าการประเมินด้วยเวลา และถ้าในครั้งแรกเราประเมินเวลาผิดไป เราจะเรียนรู้ได้ใน Iteration ถัดไป

3. ปัจจัยความสำเร็จในการเลือกใช้วิธีการพัฒนาระบบแบบเอจายล์

จากข้อมูลของ Communications of the ACM ได้ทำการสรุปว่าปัจจัยสำเร็จของการเลือกใช้วิธี Agile ต้องเกิดจากความร่วมมือ 4 ด้าน ซึ่งอธิบายได้ ดังนี้

1) ฝ่ายบริหารหรือการสนับสนุนจากองค์กร

1.1) วัฒนธรรมองค์กร

1.2) รูปแบบในการบริหารงาน

1.3) รูปแบบขององค์กร

1.4) ความรู้และความเข้าใจเรื่องการพัฒนาซอฟต์แวร์ของฝ่ายบริหาร

1.5) ระบบการให้ผลตอบแทน

2) ความพร้อมในระดับบุคคล

2.1) ประสิทธิภาพในการทำงานร่วมกันในลักษณะของทีม

2.2) ศักยภาพการทำงานในระดับสูง

2.3) วิธีการบริหารความสัมพันธ์กับลูกค้า เช่น ความเชื่อถือซึ่งกันและกัน

3) ความพร้อมในแง่กระบวนการ

3.1) ขั้นตอนการจัดการเมื่อเกิดการเปลี่ยนแปลง

3.2) การกำหนดรอบการพัฒนาหนึ่ง ๆ

3.3) การกำหนดรูปแบบเมื่อต้องบริหารโครงการขนาดใหญ่

3.4) การกำหนดใช้วิธีการย่อย ๆ ในลักษณะของ Agile

4) การมีเครื่องมือหรือเทคนิคต่าง ๆ ช่วยเหลือ

4.1) การประยุกต์ใช้เครื่องมือที่มีอยู่แล้วให้เหมาะสม

4.2) การเตรียมพร้อมรับความรู้หรือเทคนิคใหม่ ๆ อยู่ตลอดเวลา

เครื่องมือสนับสนุนกระบวนการพัฒนาระบบ

อรยา ปรีชาพานิช, (2557) เครื่องมือสนับสนุนกระบวนการพัฒนาระบบเรียกอย่างเป็นทางการว่า เครื่องมือทางวิศวกรรมซอฟต์แวร์ที่ ใช้คอมพิวเตอร์ช่วยดำเนินการ (Computer - Aided Software Engineering Tools : CASE Tools) ซึ่งถูกพัฒนา ขึ้นเพื่อใช้สนับสนุนการทำงานในแต่ละระยะของ SDLC ให้บรรลุเป้าหมายได้อย่างรวดเร็วและมีประสิทธิภาพสูง กว่าการทำงานด้วยมือ โดยสามารถสรุปวัตถุประสงค์ในการใช้งาน CASE Tools ได้ดังนี้

1. เพื่อปรับปรุงคุณภาพของระบบในภาพรวม
2. เพื่อสร้างมาตรฐานของกระบวนการพัฒนาระบบ
3. เพื่อลดระยะเวลาการออกแบบและพัฒนาระบบ
4. เพื่อลดค่าใช้จ่ายและเวลาในการทดสอบระบบและบำรุงรักษาระบบ
5. เพื่อปรับปรุงคุณภาพและความสมบูรณ์ของเอกสารประกอบการพัฒนาระบบ
6. เพื่อปรับปรุงประสิทธิภาพในการจัดการโครงการ
7. เพื่ออำนวยความสะดวกในการนำรหัสโปรแกรมกลับมาใช้ใหม่อีกครั้ง
8. เพื่อรองรับการทำงานของระบบข้ามแพลตฟอร์มและสภาพแวดล้อมในการดำเนินงานที่ต่างกันได้

1. ระดับการทำงานของ CASE Tools

อรยา ปรีชาพานิช, (2557) CASE Tools ที่ใช้สนับสนุนกระบวนการทำงานตาม SDLC จะแบ่งออกเป็น 3 ระดับ ดังต่อไปนี้

1) Upper CASETools เป็นเครื่องมือสนับสนุนกระบวนการทำงานวิเคราะห์และออกแบบระบบ โดย ผลลัพธ์ทั้งหมดจะถูกรวบรวมไว้ที่ CASE Repository ซึ่งเป็นฐานข้อมูลสำหรับจัดเก็บสารสนเทศ ของระบบ ประกอบด้วยระเบียนและส่วนย่อยของข้อมูล แผนภาพ หน้าจอ รายงาน และสารสนเทศ อื่น ๆ ที่เกี่ยวข้อง จากหลักการเก็บรวบรวมข้อมูลไว้ในแหล่งเดียวกันนี้ทำให้ CASE Tools สามารถวิเคราะห์ผลของการออกแบบระบบที่ยังมีข้อผิดพลาดหรือยังไม่สมบูรณ์ได้อย่างรวดเร็ว

2) Lower CASETools เป็นเครื่องมือสนับสนุนกระบวนการทำงานในระยะปลายของ SDLC เช่น การสร้างรหัสโปรแกรมและการสร้างกรณีทดสอบโดยอัตโนมัติ เป็นต้น

3) Integrated CASETools (I - CASE) เป็นเครื่องมือสนับสนุนกระบวนการทำงานครอบคลุมกิจกรรมทั้งหมดของ SDLC ตามที่ได้กล่าวไว้แล้วในระดับของ Upper CASE Tools และ Lower CASE Tools

เพื่อให้เข้าใจถึงหลักการประยุกต์ใช้ CASE Tools และประโยชน์ที่โครงการจะได้รับในภาพรวม เมื่อเปรียบเทียบกับการทำงานแบบดั้งเดิม นั่นคือ การประยุกต์ใช้ CASE Tools จะช่วย ลดภาระการทำงานในส่วนของการเขียนโปรแกรมและการทดสอบโปรแกรม นอกจากนั้นยังสามารถลดผลย้อนกลับ ที่เกี่ยวข้องกับข้อผิดพลาดและ Bug ของโปรแกรม รวมไปถึงปัญหาในการติดตั้งระบบได้อีกด้วย

2. ความสามารถของ CASE Tools

อรยา ปรีชาพานิช, (2557) ปัจจุบันได้มีบริษัทผู้ผลิต CASE Tools ที่หลากหลายให้นักวิเคราะห์ระบบได้เลือกใช้งาน เช่น Microsoft Visio, Oracle Designer, Rational Rose, SmartDraw และ PowerDesigner เป็นต้น โดยมีคุณสมบัติและความสามารถในการสนับสนุนการปฏิบัติงานแตกต่างกันไป

โดยทั่วไปมักจะแบ่งความสามารถของ CASE Tools ตามการสนับสนุนการปฏิบัติงานตาม SDLC ได้ดังนี้

1. เครื่องมือสร้างแบบจำลองและแผนภาพ (Modeling and Diagramming CASE Tools) เป็นเครื่องมือสนับสนุนกระบวนการทำงานในระยะของการวิเคราะห์และออกแบบระบบ สำหรับสร้าง แบบจำลองและแผนภาพหลายประเภท เพื่อแสดงความสัมพันธ์ระหว่างกิจกรรมและข้อมูล สารสนเทศในแต่ละลำดับชั้นการทำงานของระบบ โดย CASE Tools ที่มีประสิทธิภาพนั้นนอกจากจะช่วยในการสร้างแบบจำลองและแผนภาพแล้ว ยังสามารถตรวจสอบความถูกต้องและความสอดคล้องกันในแต่ละลำดับชั้น (Function Hierarchy Diagramming) ได้อีกด้วย

2. เครื่องมือสร้างโปรแกรมต้นแบบ (Prototyping Generators) เป็นเครื่องมือสนับสนุนกระบวนการ สร้างโปรแกรมต้นแบบ โดยแปลงความต้องการใช้งานที่ผ่านการวิเคราะห์แล้วไปสร้างฐานข้อมูลและการออกแบบระบบในเบื้องต้น รวมไปถึงการสร้างแบบฟอร์ม/รายงานต่าง ๆ ที่เรียกดูได้จากหน้าจอ หรือสั่งพิมพ์ให้อยู่ในรูปแบบของกระดาษ เพื่อใช้เป็นโปรแกรมต้นแบบสำหรับให้ผู้ใช้ได้ตรวจสอบความครบถ้วนสมบูรณ์ของระบบ นอกจากนั้นแล้วยังสามารถกำหนดรูปแบบมาตรฐานของโปรแกรมต้นแบบ เช่น ส่วนหัวและส่วนท้ายของหน้าจอรวมถึงแบบฟอร์ม/รายงานต่าง ๆ ที่สามารถเรียกใช้ซ้ำจากต้นแบบที่กำหนดไว้ล่วงหน้าได้โดยอัตโนมัติ

3. เครื่องมือสำหรับการวิเคราะห์ (Analysis CASE Tools) เป็นเครื่องมือสำหรับใช้ตรวจสอบความถูกต้องและความสอดคล้องกันของข้อกำหนดคุณลักษณะเฉพาะ (Specification) กับแผนภาพแบบจำลองหน้าจอ และแบบฟอร์ม/รายงานต่าง ๆ

4. เครื่องมือสร้างเอกสาร (DocumentationGenerators) เป็นเครื่องมือสำหรับสร้างเอกสารประกอบ การทำงานในแต่ละระยะของ SDLC จากสารสนเทศที่จัดเก็บไว้ใน CASE Repository คุณสมบัติสำคัญของเครื่องมือนี้ คือ การสร้างเอกสารแม่แบบ (Template) ที่เชื่อมโยงสารสนเทศที่สำคัญของการทำงานแต่ละระยะ รวมไปถึงการสร้างรายการของประเด็นหลักที่ต้องตรวจสอบความถูกต้องในแต่ละระยะของ SDLC ด้วย

5. เครื่องมือสร้างรหัสโปรแกรม (Code Generators) เป็นเครื่องมือสำหรับสร้างรหัสโปรแกรมจากแผนภาพ แบบจำลอง แบบฟอร์ม รายงาน และคำอธิบายข้อมูลต่าง ๆ ที่จัดเก็บไว้ใน CASE Repository ซึ่งนักวิเคราะห์ระบบสามารถกำหนดแพลตฟอร์มที่จะใช้ในการพัฒนาและติดตั้งใช้งานระบบ เนื่องจาก CASE Tools ประเภทนี้ถูกสร้างให้มีความยืดหยุ่นในการคอมไพล์และแสดงผลลัพธ์ ได้กับฮาร์ดแวร์และระบบปฏิบัติการที่หลากหลายได้

นอกจากนั้นแล้ว CASE Tools ที่มีประสิทธิภาพยังสนับสนุนการดำเนินการด้าน Reverse- Engineering เพื่อย้อนกลับไปสร้างข้อกำหนดของการออกแบบระบบ รวมทั้งแบบจำลองต่าง ๆ จากรหัสโปรแกรมและโครงสร้างข้อมูลที่มีอยู่อีกด้วย

3. มุมมองของการใช้ CASE Tools

อรยา ปรีชาพานิช, (2557) การใช้งาน CASE Tools เพื่อสนับสนุนการพัฒนาระบบมีข้อดี ทั้งในเรื่องของมาตรฐานในการพัฒนา ระบบ การลดระยะเวลาในการดำเนินงานตั้งแต่ขั้นตอนของการวิเคราะห์และออกแบบระบบ การสร้างโปรแกรม ต้นแบบ การพัฒนาและทดสอบระบบรวมไปถึงการบำรุงรักษาระบบ

นอกจากนั้นแล้วยังสามารถเพิ่มคุณภาพของระบบในภาพรวม เนื่องจาก CASE Tools มีฟังก์ชันสำหรับการตรวจสอบความถูกต้องและความสอดคล้องกันของผลผลิตในแต่ละช่วงเวลา พร้อมทั้งฟังก์ชันสำหรับสร้างเอกสารประกอบการดำเนินงานที่สามารถปรับปรุงให้สอดคล้องกับระบบที่พัฒนาขึ้นได้ตลอดเวลา

อย่างไรก็ตามการตัดสินใจใช้งาน CASE Tools ย่อมส่งผลกระทบต่อองค์กรในส่วนของค่าใช้จ่ายที่เพิ่มขึ้น เพราะ CASE Tools ยิ่งมีประสิทธิภาพมากขึ้นเท่าไร ค่าใช้จ่ายที่ตามมาก็สูงขึ้นเท่านั้น ทั้งค่าใช้จ่ายในการคำนวณจากจำนวนเครื่องที่ติดตั้งใช้งาน ค่าฝึกอบรมในการใช้งาน ค่าบำรุงรักษาระบบ และค่าปรับปรุงเวอร์ชันของ CASE Tools รวมไปถึงทัศนคติในการทำงานและการจัดการกระบวนการพัฒนาระบบในองค์กรให้เกิดมาตรฐานร่วมกัน

จากที่กล่าวมาแล้วข้างต้นจะเห็นได้ว่า การตัดสินใจใช้งาน CASE Tools จะต้องมีการประเมินถึงทรัพยากรที่ใช้ในการลงทุนและผลที่ได้จากการลงทุนให้รอบคอบเพื่อให้สามารถใช้งาน CASE Tools อย่างคุ้มค่าที่สุด

สรุปท้ายบทที่ 2

วงจรการพัฒนาระบบ (System Development Life Cycle: SDLC) มี 7 ขั้นตอนหลัก ได้แก่

1. การกำหนดความต้องการของระบบ
2. การวิเคราะห์ระบบ
3. การออกแบบระบบ
4. การพัฒนาระบบ
5. การทดสอบระบบ
6. การติดตั้งระบบ
7. การบำรุงรักษาระบบ

แบบจำลองของวงจรการพัฒนาระบบ (System Development Life CycleModel : SDLC Model) มีหลายรูปแบบ ได้แก่

1. แบบจำลองน้ำตก
2. แบบจำลองปรับปรุงน้ำตก
3. แบบจำลองแบบเพิ่มผลลัพธ์
4. แบบจำลองแบบเกลียว
5. แบบจำลองการสร้างโปรแกรมต้นแบบในระยะเริ่มต้น

ระเบียบวิธีการพัฒนาระบบแบบดั้งเดิม (Traditional System Development Methodology) มีหลายวิธี ได้แก่

1. ระเบียบวิธีการพัฒนาระบบเชิงโครงสร้าง
2. ระเบียบวิธีการพัฒนาโปรแกรมประยุกต์แบบรวดเร็ว
3. ระเบียบวิธีการพัฒนาโปรแกรมประยุกต์แบบร่วมกัน
4. ระเบียบวิธีการพัฒนาระบบแบบ Rational Unified

5 ระเบียบวิธีการพัฒนาระบบแบบเชิงวัตถุ

ระเบียบวิธีการพัฒนาซอฟต์แวร์แบบเอจายล์ (Agile Software Development Methodology) ปัจจุบันมีกระบวนการพัฒนาแบบ Agile Model หลายรูปแบบ ได้แก่

1. วิธีการพัฒนาแบบ Extreme Programming (XP)
2. วิธีการพัฒนาแบบ Scrum
3. วิธีการพัฒนาซอฟต์แวร์แบบปรับตัว
4. วิธีการพัฒนาระบบแบบไม่หยุดนิ่ง

5. วิธีการพัฒนาแบบคริสตัล
6. วิธีการพัฒนาที่ขับเคลื่อนด้วยคุณลักษณะของซอฟต์แวร์

เครื่องมือสนับสนุนกระบวนการพัฒนาระบบ คือ เครื่องมือสนับสนุนกระบวนการพัฒนาระบบเรียกอย่างเป็นทางการว่า เครื่องมือทางวิศวกรรมซอฟต์แวร์ที่ ใช้คอมพิวเตอร์ช่วยดำเนินการ (Computer - Aided Software Engineering Tools : CASE Tools) ซึ่งถูกพัฒนา ขึ้นเพื่อใช้สนับสนุนการทำงานในแต่ละระยะของ SDLC ให้บรรลุเป้าหมายได้อย่างรวดเร็วและมีประสิทธิภาพสูง แบ่งเป็น 3 หัวข้อหลัก ดังนี้

1. ระดับการทำงานของ CASE Tools
2. ความสามารถของ CASE Tools
3. มุมมองของการใช้ CASE Tools

คำถามท้ายบทที่ 2

1. อธิบายขั้นตอนการดำเนินงานของวงจรการพัฒนาระบบ พร้อมทั้งระบุกิจกรรมย่อยและผลลัพธ์ที่ได้ จากการดำเนินงานในแต่ละขั้นตอน
2. ยกตัวอย่างแบบจำลองของวงจรการพัฒนาระบบ 2 รูปแบบ โดยวาดภาพแบบจำลอง พร้อมทั้งอธิบายรายละเอียดให้เข้าใจถึงการทำงานในภาพรวม
3. อธิบายความหมายและองค์ประกอบของระเบียบวิธีการพัฒนาระบบ พร้อมทั้งยกตัวอย่างประกอบ
4. อธิบายความแตกต่างระหว่างระเบียบวิธี Structured Analysis and Design (SAD) และ Rapid Application Development (RAD) พร้อมทั้งสรุปข้อดีและข้อจำกัดของแต่ละระเบียบวิธี
5. อธิบายคุณลักษณะสำคัญของระเบียบวิธี Agile พร้อมทั้งสรุปข้อดีและข้อจำกัดของระเบียบวิธีนี้
6. อธิบายความสามารถของ CASE Tools ในด้านต่าง ๆ มา 3 ด้านพอสังเขป

บทที่ 3
การวิเคราะห์ความต้องการของผู้ใช้

การวิเคราะห์ความต้องการของผู้ใช้ คือ การค้นหาและเข้าใจถึงความต้องการและความคาดหวังของผู้ใช้ที่มีต่อระบบที่กำลังจะพัฒนา ซึ่งเป็นส่วนหนึ่งของวงจรการพัฒนาระบบ (System Development Life Cycle : SDLC) หลังจากการดำเนินการสำรวจเบื้องต้น ขั้นตอนการวิเคราะห์นี้มุ่งเน้นไปที่การรวบรวมข้อมูลจากผู้ใช้งานทุกฝ่ายที่เกี่ยวข้องกับโครงการ และต่อมาจะวิเคราะห์ข้อมูลเหล่านั้นเพื่อสร้างเป็นความต้องการของระบบที่ชัดเจนและเป็นมืออาชีพ

การวิเคราะห์ระบบไม่ใช่เพียงแค่ขั้นตอนที่เกิดขึ้นในวงจรการพัฒนาระบบ แต่ยังเป็นกิจกรรมที่มีความสำคัญในการกำหนดทิศทางของโครงการ ผู้ที่มีส่วนเกี่ยวข้องในโครงการไม่ว่าจะเป็นนักพัฒนาหรือผู้ใช้งานควรมีความเข้าใจเกี่ยวกับความสำคัญของการวิเคราะห์ระบบ และควรทำงานร่วมกันในการกำหนดความต้องการและความคาดหวังของระบบเพื่อให้การพัฒนาระบบที่จะเกิดขึ้นต่อไปในขั้นตอนต่างๆ ของ SDLC มีประสิทธิภาพ การวิเคราะห์ความต้องการของผู้ใช้จึงเป็นสิ่งที่จำเป็นอย่างยิ่ง ด้วยการวิเคราะห์ที่ถูกต้องและครอบคลุม จะทำให้การพัฒนาระบบต่อไปเป็นไปอย่างราบรื่นและตอบสนองต่อความต้องการของผู้ใช้งานได้อย่างแท้จริง ในบทนี้จึงแบ่งเนื้อหาออกได้ ดังนี้

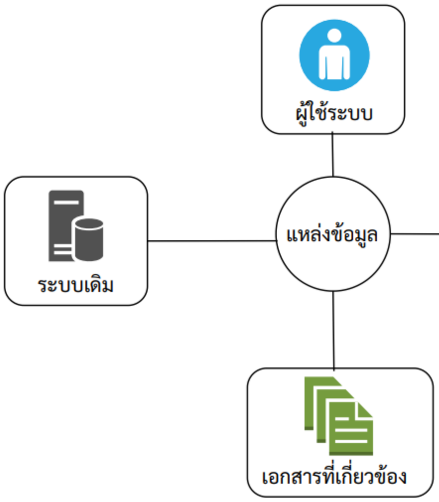
การรวบรวมความต้องการของผู้ใช้งาน

การพัฒนาระบบสารสนเทศในยุคปัจจุบันเป็นกระบวนการที่ต้องคำนึงถึงความต้องการของผู้ใช้งานในทุก ๆ ฝ่าย ไม่ว่าจะเป็นผู้บริหาร หัวหน้าฝ่าย หรือเจ้าหน้าที่ภายในองค์กร แต่ละบุคคลมีวิสัยทัศน์และความต้องการที่แตกต่างกัน ซึ่งทำให้การรวบรวมข้อมูลความต้องการ (User Requirement Gathering) กลายเป็นกิจกรรมที่มีความซับซ้อนและเป็นท้าทาย การวิเคราะห์ระบบเป็นกิจกรรมที่ต้องการความรู้ความเข้าใจในด้านจิตวิทยาเพื่อให้สามารถเข้าถึงความคิดและความต้องการของผู้ใช้งานได้อย่างถูกต้อง ในขั้นตอนนี้สามารถเปรียบเสมือนกับภูเขาน้ำแข็งที่เราเห็นเพียงส่วนยอดเท่านั้น แต่ใต้น้ำยังมีส่วนที่กว้างขวางและซับซ้อนกว่าที่เราเห็น นักวิเคราะห์ระบบจึงต้องมีความแสวงหาและอดทนเพื่อสามารถเจาะลึกและเข้าถึงความต้องการที่ซ่อนเร้นของผู้ใช้งาน แนวคิดในการวิเคราะห์ความต้องการของผู้ใช้งานเป็นสิ่งที่สำคัญในการพัฒนาระบบ หากเราสามารถรวบรวมความต้องการของผู้ใช้งานได้อย่างครบถ้วน การพัฒนาระบบในอนาคตก็จะสามารถตอบสนองความต้องการของผู้ใช้งานได้อย่างแม่นยำ และเป็นที่ยอมรับของผู้ใช้งาน

ดังนั้น นักวิเคราะห์ระบบยังต้องพิจารณาถึงความสมบูรณ์และความถูกต้องของข้อมูลที่ได้รับ การทำให้แน่ใจว่าความต้องการที่ได้รวบรวมมานั้นครบถ้วนและถูกต้องจะช่วยลดความเสี่ยงที่จะเกิดขึ้นในการพัฒนาระบบในขั้นตอนต่อไปของ SDLC การวิเคราะห์ระบบเป็นกิจกรรมที่ต้องการความร่วมมือจากทุกฝ่ายที่เกี่ยวข้อง การสื่อสารและความเข้าใจร่วมกันระหว่างนักวิเคราะห์ระบบและผู้ใช้งานเป็นสิ่งที่สำคัญในการทำให้การพัฒนาระบบสำเร็จลุล่วง

1. แหล่งข้อมูลของความต้องการใช้งานระบบ

แหล่งข้อมูลของความต้องการใช้งานระบบที่นักวิเคราะห์ระบบจะต้องศึกษา เพื่อทำความเข้าใจและเก็บรวบรวมข้อมูลมีหลายแหล่ง ซึ่งแต่ละแหล่งอาจจะให้ข้อมูลและสารสนเทศที่แตกต่างกัน ดังนั้น การเข้าถึงข้อมูลให้ครบทุกแหล่งจึงช่วยให้นักวิเคราะห์ระบบเกิดความเข้าใจและได้ข้อสรุปของระบบใหม่ที่จะพัฒนาได้อย่างถูกต้อง แหล่งข้อมูลมีทั้งจากภายในองค์กรและภายนอกองค์กร สามารถแบ่งเป็น 4 กลุ่มหลัก (อรยา ปรีชาพานิช, 2557) ดังภาพ 3.1



ภาพที่ 3.1 แหล่งข้อมูลของความต้องการของระบบ

จากภาพที่ 3.1 แหล่งข้อมูลของความต้องการของระบบ สามารถอธิบายแต่ละแหล่งข้อมูล ดังนี้

1.1 ผู้ใช้ระบบ

ผู้ใช้ระบบคือกลุ่มหลักที่เกี่ยวข้องกับการพัฒนาระบบใหม่ และเป็นแหล่งข้อมูลหลักในการรวบรวมความต้องการ เนื่องจากเป็นผู้ที่มีส่วนร่วมโดยตรงในการปฏิบัติงานที่เกี่ยวข้องกับระบบ การสื่อสารและเข้าใจความต้องการของพวกเขาเป็นสิ่งสำคัญ นักวิเคราะห์ระบบจึงควรใช้เวลาในการศึกษาและค้นหาข้อมูลจากผู้ใช้เพื่อทำความเข้าใจกระบวนการทางธุรกิจและความต้องการที่ผู้ใช้คาดหวังจากระบบใหม่

1.2 ผู้บริหาร

ผู้บริหารเป็นกลุ่มที่มีอิทธิพลต่อนโยบายและทิศทางของระบบ ผู้บริหารมีวิสัยทัศน์เกี่ยวกับการใช้ระบบในการสนับสนุนการตัดสินใจ การจัดการ และการควบคุม การศึกษาและเข้าใจความต้องการของผู้บริหารจะช่วยในการกำหนดความสามารถและขอบเขตสำคัญของระบบ

1.3 ระบบเดิมที่ใช้ในปัจจุบัน

การศึกษาระบบที่ใช้อยู่ในปัจจุบันเป็นส่วนสำคัญของการวิเคราะห์ระบบ การทำความเข้าใจวิธีการทำงาน กระบวนการ และผลลัพธ์ของระบบปัจจุบันจะช่วยในการระบุข้อจำกัด โอกาส และปัญหาที่อาจเกิดขึ้นในระบบใหม่

1.4 เอกสารที่เกี่ยวข้อง

เอกสารต่างๆ เป็นแหล่งข้อมูลที่มีค่าในการวิเคราะห์ระบบ นอกจากคู่มือการปฏิบัติงาน แบบฟอร์ม และรายงาน นักวิเคราะห์ระบบยังควรพิจารณาเอกสารอื่น ๆ เช่น นโยบาย แผนภูมิโครงสร้างองค์กร

รายงานประจำปี และกฎระเบียบที่เกี่ยวข้องเพื่อให้ได้ภาพรวมที่ครบถ้วนของความต้องการและความคาดหวังขององค์กร

1) เอกสารประเภทคู่มือการปฏิบัติงาน เอกสารประเภทนี้เป็นแหล่งข้อมูลที่บรรยายเกี่ยวกับหน้าที่และความรับผิดชอบของตำแหน่งต่าง ๆ ภายในองค์กร บางครั้งอาจประกอบไปด้วยแผนภาพกระแสงาน (Work Flow Diagram) เพื่อสื่อสารลำดับการทำงานและเพื่อให้พนักงานมีความเข้าใจในกระบวนการทำงานขององค์กร โครงสร้างองค์กรและระเบียบ/ข้อบังคับต่าง ๆ ขององค์กร

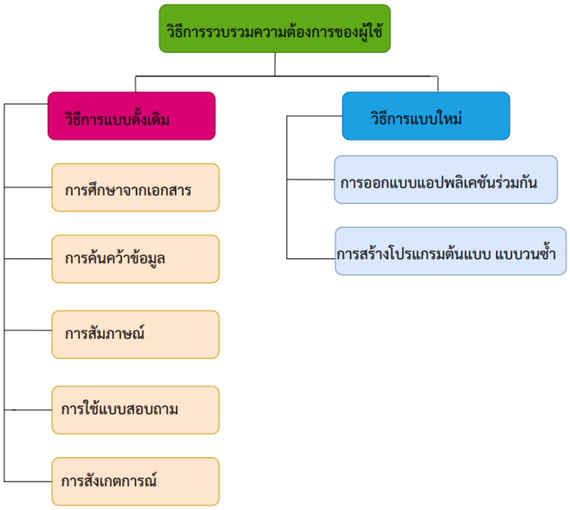
2) แบบฟอร์มและรายงานผลการปฏิบัติงาน เป็นแหล่งข้อมูลพื้นฐานสำหรับนักวิเคราะห์ระบบเพื่อกำหนดโครงสร้างข้อมูล แบบฟอร์มในการขายสินค้า เช่น ใบสั่งขาย ใบเบิกสินค้า ใบส่งของ ใบแจ้งหนี้และใบเสร็จรับเงิน เป็นต้น ขณะเดียวกัน รายงานต่าง ๆ เช่น รายงานยอดขายประจำวัน/เดือน/ปี รายงานสรุปสินค้าขายดี รายงานสินค้าที่ยอดขายต่ำกว่าเป้าหมาย เป็นต้น

3) เอกสารในภาพรวมขององค์กร เอกสารเหล่านี้ เช่น นโยบาย พันธกิจ แผนภูมิโครงสร้าง รายงานประจำปี รายงานการเงิน รายงานสถิติ และรายงานการประชุม เป็นแหล่งข้อมูลที่มีความสำคัญในการวิเคราะห์ระบบเพื่อให้สอดคล้องกับวิสัยทัศน์และเป้าหมายขององค์กร

4) เอกสารจากภายนอกองค์กร เป็นเอกสารซึ่งรวมถึงเอกสารความรู้ กฎระเบียบของราชการ และรายงานของสมาคมหรือองค์กรที่ทำงานในด้านเดียวกัน เป็นแหล่งข้อมูลเสริมที่สามารถช่วยในการวิเคราะห์ระบบในมุมมองที่แตกต่างและเสริมสร้างความเข้าใจในภาพรวม

2. วิธีการรวบรวมความต้องการของผู้ใช้

วิธีการรวบรวมความต้องการของผู้ใช้ (User Requirement Gathering Method) สามารถแบ่งได้เป็น 2 กลุ่มหลัก คือ วิธีการรวบรวมความต้องการของผู้ใช้แบบดั้งเดิม (Traditional User Requirement Gathering Method) และวิธีการรวบรวมความต้องการของผู้ใช้แบบใหม่ (Modern User Requirement Gathering Method) (อรยา ปรีชาพานิช, 2557) ดังภาพที่ 3.2



ภาพที่ 3.2 วิธีการรวบรวมความต้องการของผู้ใช้

2.1 การศึกษาจากเอกสาร

การวิเคราะห์ข้อมูลผ่านเอกสารที่เกี่ยวข้อง (Document Review) คือ วิธีการที่ทั้งเป็นมาตรฐานและไม่ยุ่งยากในการรวบรวมความต้องการของผู้ใช้งานระบบภายในองค์กรที่จะพัฒนาระบบ นักวิเคราะห์ระบบสามารถสอบถามเอกสารที่เกี่ยวข้องโดยตรงเพื่อให้ทราบถึงความต้องการและรายละเอียดของระบบ เอกสารต่าง ๆ เช่น คู่มือการปฏิบัติงาน แบบฟอร์ม หรือรายงานที่เกี่ยวข้องสามารถให้ข้อมูลที่มีประโยชน์ต่อการวิเคราะห์ แนะนำให้นักวิเคราะห์ระบบร้องขอเอกสารที่มีข้อมูลตัวอย่างภายใน เช่น แบบฟอร์มที่เติมข้อมูลแล้ว หรือรายงานที่มีค่าสถิติ เพื่อทำให้ได้มุมมองที่ครบถ้วนของกระบวนการทำงานและข้อมูลที่ใช้ในระบบ การมีเอกสารประกอบการวิเคราะห์เช่นนี้จะช่วยในการรวบรวมและวิเคราะห์ข้อมูลในรายละเอียดและความลึกอย่างมีประสิทธิภาพ

2.2 การค้นคว้าข้อมูล

ในปัจจุบันเครือข่ายอินเทอร์เน็ตเป็นแหล่งข้อมูลขนาดใหญ่ที่สำคัญ ทำให้นักวิเคราะห์ระบบสามารถดำเนินการค้นคว้าข้อมูล (Research) อย่างมีประสิทธิภาพและรวดเร็วกว่าที่เคยเป็นมา การศึกษาจากเอกสารที่ได้รับจากหน่วยงานต่าง ๆ เป็นแหล่งข้อมูลที่สำคัญสำหรับการเสริมสร้างความรู้เพิ่มเติม เช่น เอกสารที่เผยแพร่ งานวิจัย หนังสือ นิตยสาร และวารสาร การดึงข้อมูลจากแหล่งเหล่านี้ช่วยให้นักวิเคราะห์ระบบสามารถเริ่มกระบวนการรวบรวมข้อมูลที่ต้องการใช้งานจากแหล่งข้อมูลภายนอกได้อย่างมีประสิทธิภาพ เป็นการก่อนหน้า

การขอเอกสารจากหน่วยงานต่าง ๆ โดยเฉพาะอย่างยิ่งในกรณีที่หน่วยงานนั้นมีเว็บไซต์ของตนที่เปิดเผยข้อมูลพื้นฐาน ที่นักวิเคราะห์ระบบสามารถเข้าถึงได้อย่างสะดวกและรวดเร็ว

วิธีการนี้เป็นกระบวนการหนึ่งที่นักวิเคราะห์ระบบใช้เพื่อเตรียมข้อมูลสำคัญสำหรับการวิเคราะห์และการวิจัย การค้นคว้าข้อมูลที่เกี่ยวข้องจากแหล่งเอกสารเป็นขั้นตอนสำคัญในการก้าวสู่ขั้นตอนถัดไปของกระบวนการดังกล่าว อย่างไรก็ตาม การขอเอกสารจากหน่วยงานยังคงเป็นกระบวนการที่จำเป็น โดยเฉพาะเมื่อต้องการข้อมูลที่ไม่เผยแพร่หรือเข้าถึงได้ยาก หากหน่วยงานมีเว็บไซต์ออนไลน์ที่ให้ข้อมูลพื้นฐาน เป็นที่สามารถเข้าถึงได้อย่างสะดวก นักวิเคราะห์ระบบจะมีความสะดวกในการเข้าถึงแหล่งข้อมูลเพื่อสนับสนุนงานวิจัยและการวิเคราะห์อย่างมีประสิทธิภาพมากยิ่งขึ้น การรวบรวมข้อมูลที่มีคุณภาพและถูกต้องเป็นการสร้างพื้นฐานที่เป็นประโยชน์สำหรับการทำงานทางวิชาการในปัจจุบัน

2.3 การสัมภาษณ์

กระบวนการสัมภาษณ์เป็นขั้นตอนที่ธรรมดาที่เกิดขึ้นหลังจากการที่นักวิเคราะห์ระบบได้ดำเนินการศึกษาและค้นคว้าข้อมูลเกี่ยวกับหัวข้อที่เกี่ยวข้องไปแล้ว ดังนั้นการรู้และเข้าใจเกี่ยวกับระบบใหม่ที่กำลังจะถูกพัฒนาเป็นสิ่งสำคัญในการประเมินความเหมาะสมและความต้องการของระบบนั้น นักวิเคราะห์ระบบจำเป็นต้องมีความรู้และความเข้าใจเชิงลึกเกี่ยวกับคำถามที่เกี่ยวข้องกับระบบนี้ในระดับหนึ่ง เพื่อให้สามารถดำเนินกระบวนการสัมภาษณ์ได้อย่างมีประสิทธิภาพและมีความสำเร็จ

นักวิเคราะห์ระบบจำเป็นต้องรู้จักบริหารเวลาและข้อความในกระบวนการสัมภาษณ์อย่างมีประสิทธิภาพ เพื่อให้สามารถสร้างความเข้าใจที่ลึกซึ้งและเป็นรายละเอียดเกี่ยวกับแง่มุมต่าง ๆ ของระบบใหม่ที่กำลังพัฒนาขึ้น เป็นขั้นตอนที่สำคัญในกระบวนการวิเคราะห์และการพัฒนาระบบที่เน้นการรับฟังและการเข้าใจปัญหาและความต้องการของผู้เกี่ยวข้อง การสัมภาษณ์เป็นเครื่องมือที่มีประสิทธิภาพในการสกัดข้อมูลและเชิงลึกเข้าสู่ความคิดเห็นและความรู้สึกของผู้ที่มีส่วนเกี่ยวข้องกับระบบนั้น หากนำมาประยุกต์ใช้ในการวิเคราะห์ระบบและการวางแผนพัฒนา จะช่วยเพิ่มความเข้าใจและความสำเร็จในการสร้างระบบใหม่อย่างมีประสิทธิภาพมากยิ่งขึ้น

2.3.1 รูปแบบการสัมภาษณ์

ก่อนการสัมภาษณ์นักวิเคราะห์ระบบจะต้องเลือกรูปแบบของการสัมภาษณ์ที่เหมาะสมเพื่อให้ ได้รับข้อมูลที่ครบถ้วนและถูกต้องมากที่สุด โดยส่วนใหญ่มักตัดสินใจเลือกจากประสบการณ์และความถนัดของ นักวิเคราะห์ระบบเอง รวมทั้งการพิจารณาถึงวัฒนธรรมขององค์กรนั้น ๆ การสัมภาษณ์จะแบ่งเป็น 2 รูปแบบดังนี้

1) การสัมภาษณ์แบบมีโครงสร้าง (Structured Interview) เป็นการสัมภาษณ์ที่นักวิเคราะห์ระบบ จะต้องกำหนดประเด็นการสัมภาษณ์และคำถามต่าง ๆ ไว้ล่วงหน้า แล้วประสานไปยังผู้ที่

รับผิดชอบ โดยตรง เพื่อขอนัดสัมภาษณ์ในประเด็นดังกล่าว การสัมภาษณ์ในรูปแบบนี้จะมีทั้งข้อดีและข้อจำกัด ดังนี้

1.1) ข้อดี

1.1.1) นักวิเคราะห์ระบบและผู้ถูกสัมภาษณ์ต่างก็ได้เตรียมตัวมาแล้วล่วงหน้า จึงมักสามารถสื่อสารกันได้ในเวลาที่กำหนดไว้

1.1.2) การสัมภาษณ์มักจะได้รายละเอียด รวมทั้งเอกสารตัวอย่างเพิ่มเติมจากการจัดเตรียมไว้ล่วงหน้า ของผู้ถูกสัมภาษณ์

1.2) ข้อจำกัด

1.2.1) บรรยากาศการสัมภาษณ์จะค่อนข้างเป็นทางการ โดยมีการสนทนากันตามลำดับของประเด็น คำถามที่ได้ประสานงานไว้แล้วล่วงหน้า

1.2.2) การสัมภาษณ์มักต้องยึดถือตามประเด็นคำถามที่ส่งไปให้ล่วงหน้าแล้วเท่านั้นทำให้ขาดความ ยืดหยุ่นในการสัมภาษณ์นอกเหนือจากประเด็นดังกล่าว เนื่องจากผู้ใช้จะรู้สึกว่าไม่ได้เตรียมตัวในการตอบคำถามเหล่านั้น และมักร้องขอให้นัดสัมภาษณ์เพิ่มเติมในครั้งถัดไป

2) การสัมภาษณ์แบบไม่มีโครงสร้าง (Unstructured Interview) เป็นการสัมภาษณ์ที่นักวิเคราะห์ระบบอาจกำหนดประเด็นการสัมภาษณ์ไว้คร่าว ๆ เพื่อขอนัดสัมภาษณ์ในลักษณะของการสนทนาและซักถามต่อเนื่องไปตามลำดับเหตุการณ์ และเมื่อมีประเด็นที่น่าสนใจก็อาจถามคำถามที่เจาะลึก ณ ขณะนั้นได้ทันที การสัมภาษณ์ในรูปแบบนี้จะมีทั้งข้อดีและข้อจำกัด ดังนี้

2.1) ข้อดี

2.1.1) บรรยากาศการสัมภาษณ์จะค่อนข้างเป็นกันเอง โดยใช้วิธีการสนทนาแลกเปลี่ยนความคิดเห็นซึ่งไม่เป็นทางการ

2.1.2) การสัมภาษณ์มีความยืดหยุ่นสูงมาก เพราะทั้งผู้สัมภาษณ์และผู้ถูกสัมภาษณ์จะไม่มีประเด็น คำถามที่กำหนดไว้ล่วงหน้า

2.2) ข้อจำกัด

2.2.1) การควบคุมเวลาในการสัมภาษณ์จะเป็นไปได้ยากกว่าการสัมภาษณ์แบบมีโครงสร้าง

2.2.2) การสรุปและประเมินผลการสัมภาษณ์เป็นไปได้ยากกว่าการสัมภาษณ์แบบมีโครงสร้าง

2.3.2 ขั้นตอนการสัมภาษณ์

เมื่อเลือกรูปแบบของการสัมภาษณ์แล้ว นักวิเคราะห์ระบบจะดำเนินการตามขั้นตอนการสัมภาษณ์ ซึ่งประกอบด้วยรายละเอียดการดำเนินงาน

1) กำหนดตัวบุคคลที่จะสัมภาษณ์

โดยคำนึงถึงประเด็นในการสัมภาษณ์แต่ละครั้งเป็นหลัก ในขั้นตอนนี้ นักวิเคราะห์ระบบควรประสานงานผ่านผู้ที่รับผิดชอบโครงการนี้โดยตรง เพื่อให้ช่วยคัดเลือก และตรวจสอบความเหมาะสมของบุคคลที่จะถูกสัมภาษณ์ในประเด็นนั้น ๆ ในการสัมภาษณ์แต่ละครั้งสามารถเชิญผู้ถูกสัมภาษณ์ได้หลายคน แต่ถ้ามีจำนวนคนมาก เกินไปนักวิเคราะห์ระบบอาจจำเป็นต้องจำกัดประเด็นการสนทนาให้แคบลง เพื่อให้การสัมภาษณ์มีประสิทธิภาพมากที่สุด

ข้อผิดพลาดที่อาจเกิดขึ้นและสร้างความเสียหายแก่โครงการได้คือ การเลือกสัมภาษณ์บุคคลที่ไม่สามารถให้ข้อมูลที่ตรงตามวัตถุประสงค์ที่กำหนดไว้ เนื่องจากการนัดสัมภาษณ์แต่ละครั้งมักใช้ระยะเวลาขั้นต่ำประมาณ 2 ชั่วโมง ดังนั้น ถ้าบุคคลที่ถูกสัมภาษณ์ไม่ใช่ผู้เชี่ยวชาญที่รู้งานดังกล่าวจริง ก็เท่ากับเสียเวลาไปโดยเปล่าประโยชน์ และอาจสร้างความเสียหายอย่างมากในกรณีที่นักวิเคราะห์ระบบหลงเชื่อข้อมูลที่ผิด ๆ และนำไปสรุปเพื่อวิเคราะห์เป็นความต้องการของระบบ

2) เตรียมการสัมภาษณ์

นักวิเคราะห์ระบบจะต้องกำหนดวัตถุประสงค์ในการสัมภาษณ์แต่ละครั้งให้ชัดเจน เพื่อใช้ในการควบคุมทิศทางการสัมภาษณ์ให้เป็นไปตามวัตถุประสงค์นั้น ๆ เพราะในบางครั้ง ผู้ถูกสัมภาษณ์อาจมีความรู้สึกต่อต้าน หรือความรู้สึกในแง่ลบกับการเปลี่ยนแปลงที่กำลังจะเกิดขึ้น ทำให้ขาดความร่วมมือในการให้สัมภาษณ์ หรือไม่ก็อาจเป็นผู้ที่มีอัธยาศัยดีสามารถเล่าเรื่องต่าง ๆ ได้มากมายแต่ไม่ตรงประเด็น ซึ่งจะทำให้เสียเวลาไปโดยใช่เหตุ ดังนั้น การสัมภาษณ์ที่ดีจะต้องกระชับ ได้ใจความและชัดเจนในประเด็นที่จะสัมภาษณ์ทั้ง ในส่วนของนักวิเคราะห์ระบบและผู้ถูกสัมภาษณ์ เพื่อให้เกิดบรรยากาศที่ดีและมีความรู้สึกเสมือนเป็นทีมเดียวกัน

3) ดำเนินการสัมภาษณ์

เป็นขั้นตอนที่สำคัญที่สุด เพราะเป็นการรวบรวมข้อมูลแบบปฐมภูมิ ซึ่งจำเป็นต้องดำเนินการอย่างหนักแน่นและนุ่มนวลในคราวเดียวกัน เพื่อให้สามารถบรรลุวัตถุประสงค์ที่กำหนดไว้อย่างครบถ้วนและรวดเร็ว แต่ในขณะเดียวกันก็ไม่สร้างความรู้สึกไม่ดีแก่ผู้ถูกสัมภาษณ์ ซึ่งเทคนิคที่สามารถนำมาใช้ในขั้นตอนนี้คือ

3.1) การสัมภาษณ์จะเริ่มต้นจากนักวิเคราะห์ระบบได้แนะนำตนเองและอธิบายถึงวัตถุประสงค์ในการ สัมภาษณ์ พร้อมทั้งผลความสำเร็จที่จะตามมาจากการรวบรวมข้อมูลความต้องการใช้งานระบบเพื่อให้ผู้ถูกสัมภาษณ์ตระหนักถึงความสำคัญในการให้ความร่วมมือในการสัมภาษณ์

3.2) นักวิเคราะห์ระบบจะต้องสร้างบรรยากาศที่เป็นกันเอง ไม่เคร่งเครียดจนสร้างความกดดันให้แก่ผู้ถูกสัมภาษณ์ โดยเริ่มจากคำถามทั่วไปก่อนจะลงลึกในข้อมูลเกี่ยวกับระบบ

3.3) การนัดสัมภาษณ์แต่ละครั้งควรใช้เวลาที่พอเหมาะ เพื่อไม่ให้เกิดความรู้สึกน่าเบื่อหน่าย และ ส่งผลกระทบกับหน้าที่ความรับผิดชอบประจำวันของผู้ถูกสัมภาษณ์

3.4) นักวิเคราะห์ระบบควรสอดแทรกการชี้แจงให้ผู้สัมภาษณ์ตระหนักถึงความสำคัญ และความจำเป็นในการใช้ระบบคอมพิวเตอร์และเทคโนโลยีสารสนเทศ เพื่อสนับสนุนการปฏิบัติงาน

3.5) นักวิเคราะห์ระบบควรเป็นผู้ฟังที่ดี และมีการสังเกตสีหน้าและอารมณ์ของผู้ถูกสัมภาษณ์ในขณะนั้นด้วย

3.6) นักวิเคราะห์ระบบจะต้องควบคุมการสัมภาษณ์ให้อยู่ในประเด็นที่กำหนดและไม่ควรร่วมออกความเห็นหรือให้คำสัญญาใด ๆ ล่วงหน้า

3.7) นักวิเคราะห์ระบบจะต้องแยกแยะผลการให้สัมภาษณ์ให้ได้ว่า อะไรคือข้อเท็จจริง อะไรคือความคิดเห็น

3.8) นักวิเคราะห์ระบบควรใช้ภาษาที่เหมาะสมกับหน้าที่ความรับผิดชอบ และความรู้ของผู้ถูกสัมภาษณ์ เพื่อให้คำตอบที่ได้รับตรงกับความเป็นจริงที่สุด

3.9) ก่อนจบการสัมภาษณ์นักวิเคราะห์ระบบจะต้องสรุปผลการสัมภาษณ์ในภาพรวม เพื่อสร้างความเข้าใจที่ถูกต้องร่วมกัน ในระหว่างการสัมภาษณ์จะต้องมีการบันทึกข้อมูลด้วยวิธีการใดวิธีการหนึ่ง เพื่อช่วยในการจดจำข้อมูลต่าง ๆ ที่ได้จากการสัมภาษณ์

3.9.1) การจดบันทึก ในกรณีที่นักวิเคราะห์ระบบสามารถจดบันทึกข้อมูลได้อย่างรวดเร็ว โดยอาจใช้คำย่อ หรือสรุปเป็นข้อความสั้น ๆ จากนั้นเมื่อสิ้นสุดการสัมภาษณ์จึงนำบันทึกดังกล่าวมาขยายความให้ ครบถ้วนสมบูรณ์ ซึ่งวิธีการนี้ผู้ถูกสัมภาษณ์จะไม่รู้สึกอึดอัดเหมือนกับการบันทึกเสียง ข้อจำกัดของวิธีการจดบันทึกคือ การจดบันทึกจะทำให้ขาดความต่อเนื่องในการสัมภาษณ์ เพราะนักวิเคราะห์ระบบจะใช้เวลาส่วนหนึ่งไปกับการจดบันทึก ส่งผลต่อระยะเวลาในการตั้งคำถาม และบทสนทนาที่จะต้องปรับให้พอดีตามความเร็ว-ช้าของการจดบันทึกข้อมูล

3.9.2) การบันทึกเสียง ธรรมเนียมปฏิบัติที่สำคัญในการบันทึกเสียงคือ ก่อนการใช้อุปกรณ์บันทึกเสียงจะต้องมีการขออนุญาตผู้ถูกสัมภาษณ์ทุกครั้ง ซึ่งการใช้งานอุปกรณ์ดังกล่าวจะช่วยให้นักวิเคราะห์ระบบ สามารถเก็บข้อมูลได้ครบทุกข้อความ และสามารถทุ่มเทสมาธิอย่างเต็มที่ไปที่การตั้งคำถามและบท สนทนาอย่างต่อเนื่อง และสามารถสังเกตพฤติกรรมแวดล้อมไปได้พร้อม ๆ กันด้วย ข้อจำกัดของวิธีการ

บันทึกเสียงคือ ผู้ถูกสัมภาษณ์อาจจะรู้สึกอึดอัด เพราะเกิดความวิตกกังวลว่าเสียงของตนเองจะกลายเป็นหลักฐาน ในกรณีที่กล่าวถึงสถานการณ์ใด ๆ ที่เป็นเชิงลบ ทำให้ไม่กล้าแสดงความคิดเห็นตามความเป็นจริง

นอกจากนั้นนักวิเคราะห์ระบบมักจะถือว่าได้ใช้อุปกรณ์บันทึกเสียงแล้วจึงขาดความใส่ใจ ในคำตอบของผู้ถูกสัมภาษณ์ เพราะคิดว่าสามารถฟังเพื่อทบทวนคำตอบอีกครั้งได้ จนลืมคิดไปว่าถ้า อุปกรณ์เกิดการชำรุดหรือมีอุบัติเหตุเกิดขึ้น จนทำให้เสียงที่บันทึกไว้สูญหายหรือไม่ครบถ้วนก็จะเกิด ผลเสียกับโครงการมาก เพราะผู้ถูกสัมภาษณ์คงไม่ยินดีที่จะให้สัมภาษณ์ซ้ำอีกครั้ง เนื่องจากทุกคนมีงานประจำที่ต้องรับผิดชอบเช่นกัน ส่วนในประเด็นของการจัดทำรายงานสรุปผลการสัมภาษณ์ วิธีการบันทึกเสียงก็จะใช้เวลาใน การถอดเทปและสรุปให้ได้ใจความนานกว่าวิธีการจดบันทึก

4) จัดทำรายงานสรุปผลการสัมภาษณ์

หลังเสร็จสิ้นการสัมภาษณ์แต่ละครั้ง นักวิเคราะห์ระบบควรเร่งจัดรายงานสรุปผลการสัมภาษณ์ และส่งให้ผู้ถูกสัมภาษณ์ได้ตรวจสอบความถูกต้อง โดยมีการ ลงนามรับรองรายงานการประชุมไว้ในส่วนท้ายของรายงาน เพื่อใช้เป็นข้อมูลในการวิเคราะห์ความต้องการของระบบต่อไป โดยเอกสารดังกล่าวสามารถใช้เป็นหลักฐานในกรณีที่ผู้ใช้ขอเปลี่ยนแปลงความต้องการใช้งานระบบในภายหลังได้

5) ประเมินผลการสัมภาษณ์

เป็นการวิเคราะห์ข้อเท็จจริงที่ได้จากการสัมภาษณ์ว่าครบถ้วนตามที่ได้ ตั้งวัตถุประสงค์ไว้หรือไม่ และมีอคติ (Bias) มากน้อยเพียงใด ประเด็นคำถามใดที่ยังไม่แน่ใจอาจต้อง ใช้วิธีการสัมภาษณ์กับบุคคลอื่นอีกครั้ง การประเมินผลการสัมภาษณ์แต่ละครั้ง จะทำให้นักวิเคราะห์ระบบสามารถเข้าใจความต้องการใช้งานระบบมากยิ่งขึ้น และสามารถกำหนดประเด็นการสัมภาษณ์ในครั้งถัดไปได้อย่างครอบคลุมและเจาะลึกมากยิ่งขึ้นด้วย

2.3.3 ข้อควรระวังในการสัมภาษณ์

1) ผู้ใช้ส่วนใหญ่จะพิจารณาความต้องการใช้งานเฉพาะในส่วนที่ตนเองรับผิดชอบเท่านั้น และ มักจะเป็นส่วนของความต้องการที่เป็นฟังก์ชัน ซึ่งจัดได้ว่าเป็นการให้ข้อมูลในแนวดิ่ง โดยละเลยในการคิดให้ต่อเนื่องว่าการทำงานของตนเองต้องใช้ข้อมูลจากแหล่งใดบ้าง และจะต้องส่งต่อข้อมูลไปยังแหล่งใดบ้าง รวมไปถึงการคิดเชิงพัฒนาเพื่อปรับปรุงการทำงานแบบบูรณาการเพื่อลดภาระงาน ที่ซ้ำซ้อน ดังนั้น หน้าที่หนึ่งของนักวิเคราะห์ระบบคือ การกระตุ้นให้ผู้ใช้เกิดมุมมองในการพัฒนาระบบในแนวกว้างที่มีการทำงานแบบเชื่อมโยงกันเพื่อใช้ประโยชน์จากระบบคอมพิวเตอร์ที่รองรับการทำงานเหล่านั้นได้อย่างเต็มความสามารถ

2) การรวบรวมข้อมูลโดยการสัมภาษณ์ควรดำเนินการในลักษณะเป็นกลุ่มมากกว่าการสัมภาษณ์ครั้งละหนึ่งคน เพื่อใช้โอกาสนั้นในการตรวจสอบความถูกต้องของข้อมูลที่ได้รับจากหลาย ๆ แหล่งข้อมูลไปด้วยในคราวเดียวกัน

3) การตั้งคำถามที่ผิด เป็นข้อควรระวังอย่างยิ่งเพราะคำถามที่ผิดก็มีโอกาสจะนำไปสู่คำตอบที่ผิด เช่นกัน เนื่องจากนักวิเคราะห์ระบบไม่ใช่ผู้เชี่ยวชาญในศาสตร์ทุกแขนง จึงต้องพึ่งพาการรวบรวมข้อมูลจากผู้ใช้ระบบ ซึ่งถือว่าเป็นผู้ที่มีความชำนาญในงานนั้น ๆ เป็นหลัก ปัญหานี้สามารถแก้ไขได้โดยการกำหนดประเด็นการสัมภาษณ์ล่วงหน้า เพื่อเตรียมความพร้อมทั้งในส่วนของนักวิเคราะห์ระบบและผู้ให้สัมภาษณ์ นอกจากนั้นแล้วควรทบทวนคำถามที่จะใช้ในการ สัมภาษณ์ว่าไม่หลงประเด็น และไม่ใช่คำถามที่กำกวมจนกระทั่งผู้ถูกสัมภาษณ์ต้องรอฟังการขยาย ความจากนักวิเคราะห์ระบบ จึงจะเข้าใจว่าต้องการได้ข้อมูลอะไร

2.4 การใช้แบบสอบถาม

การนำเสนอแบบสอบถาม (Questionnaire) เป็นเครื่องมือที่มักถูกบูรณาการพร้อมกับกระบวนการสัมภาษณ์เพื่อเพิ่มประสิทธิภาพในการรวบรวมข้อมูล โดยการใช้แบบสอบถามนั้นจะเน้นไปที่การสะท้อนความคิดเห็นและความต้องการจากบุคคลที่เกี่ยวข้องที่จะกระจายอยู่ในหลายฝ่ายต่าง ๆ ขององค์กร เป็นพิเศษอย่างยิ่งในกรณีที่บุคคลที่เป็นผู้รับผิดชอบหลักของระบบนั้นใช้วิธีการสัมภาษณ์เพื่อรวบรวมข้อมูลรายละเอียดและความเข้าใจเกี่ยวกับประเด็นต่าง ๆ ซึ่งเป็นปัจจัยที่ส่งผลต่อการพัฒนาระบบได้อย่างมีประสิทธิภาพ

การนำเสนอแบบสอบถามมีวัตถุประสงค์ในการรวบรวมข้อมูลจากผู้ที่เกี่ยวข้องโดยตรงกับระบบ ซึ่งผู้ที่มีหน้าที่แตกต่างกันอาจมีมุมมองและความเชื่อต่างกัน การใช้แบบสอบถามทำให้สามารถเก็บรวบรวมข้อมูลจำนวนมากได้ และช่วยให้ทุกคนรู้สึกว่ามีส่วนร่วมในกระบวนการพัฒนาระบบ วิธีการนี้มีคุณสมบัติที่เป็นประโยชน์ในเรื่องการประหยัดเวลาเมื่อเทียบกับกระบวนการสัมภาษณ์ และสามารถควบคุมคำถามให้เป็นรูปแบบมาตรฐานเพื่อเป็นการเพิ่มประสิทธิภาพในการนำเสนอและวิเคราะห์ข้อมูลได้อย่างมีประสิทธิภาพ

2.4.1 รูปแบบของคำถาม คำถามที่ใช้ในแบบสอบถามมี 2 รูปแบบคือ

1) คำถามปลายเปิด (Open-ended Questions) เป็นคำถามที่สร้างขึ้นเพื่อให้ผู้ตอบแบบสอบถาม มีอิสระให้การตอบคำถาม แต่ต้องจำกัดประเด็นในการตอบคำถามให้ชัดเจน มิเช่นนั้นผู้ตอบแบบสอบถามอาจสับสนและให้ข้อมูลไม่ตรงกับความต้องการของนักวิเคราะห์ระบบ ดังนั้น คำถาม ควรใช้คำที่เข้าใจได้ง่าย เช่น ทำไม อย่างไร หรืออะไร ซึ่งมักจะมีการสอบถามถึงความคิดเห็นและ ข้อเสนอแนะเพิ่มเติมด้วย ยกตัวอย่างเช่น

1.1) ท่านคิดว่างานที่รับผิดชอบมีอุปสรรค/ปัญหาในขั้นตอนใดบ้าง และควรแก้ไขอุปสรรค/ปัญหานั้นอย่างไร

1.2) ระบบที่ใช้ในปัจจุบันเกิดปัญหาใดบ้าง

1.3) แบบฟอร์ม/รายงานที่เกี่ยวข้องกับงานที่รับผิดชอบมีอะไรบ้าง และต้องส่งแบบฟอร์ม/รายงานเหล่านั้นไปยังฝ่ายใด

1.4) ท่านมีข้อเสนอแนะในการปรับปรุงกระบวนการทำงานในปัจจุบันให้มีประสิทธิภาพมากขึ้นได้อย่างไร

2) คำถามปลายปิด (Closed-ended Questions) เป็นคำถามที่สร้างขึ้นเพื่อให้ผู้ตอบแบบสอบถามใช้เวลาสั้น ๆ ในการตอบคำถาม การสร้างคำถามปลายปิดอาจใช้วิธีการ ดังต่อไปนี้

2.1) ให้ผู้ตอบแบบสอบถามเลือกจากตัวเลือกที่กำหนดไว้แล้ว (Multiple Choices) ทั้งนี้อาจกำหนดให้เลือกข้อที่เหมาะสมมากที่สุดเพียง 1 ตัวเลือก หรือจะอนุญาตให้เลือกได้มากกว่า 1 ตัวเลือกก็ได้ขึ้นอยู่กับความเหมาะสม ยกตัวอย่างเช่น

ท่านสำเร็จการศึกษาสูงสุดในระดับใด

☐ ปริญญาเอก ☐ ปริญญาโท ☐ ปริญญาตรี

☐ อื่นๆ โปรดระบุ............

ท่านใช้งานระบบปัจจุบันโดยเฉลี่ยประมาณกี่ชั่วโมงต่อวัน

☐ 0 - 2 ชั่วโมง ☐ 3 - 5 ชั่วโมง ☐ 6 - 8 ชั่วโมง ☐ มากกว่า 8 ชั่วโมง

ในปัจจุบันงานที่ท่านรับผิดชอบจำเป็นต้องใช้งานโปรแกรมสนับสนุนการปฏิบัติงานใดบ้าง (เลือกได้มากกว่า 1 ตัวเลือก)

☐ โปรแกรมประมวลผลคำ (Word Processing)

☐ โปรแกรมแผ่นงานคำนวณ (Spreadsheet)

☐ โปรแกรมนำเสนอข้อมูล (Presentation Graphics)

☐ โปรแกรมวิเคราะห์ทางสถิติ (Statistical Analysis)

☐ อื่นๆ โปรดระบุ............

2.2) ให้ผู้ตอบแบบสอบถามประเมินระดับความคิดเห็นในแต่ละประเด็นว่ามากน้อยเพียงใด (Rating Question) ยกตัวอย่างเช่น

ท่านมีความพึงพอใจในการใช้งานระบบเดิม มากน้อยเพียงใด ☐ มากที่สุด ☐ มาก ☐ ปานกลาง ☐ น้อย ☐ น้อยที่สุด

2.3) ให้ผู้ตอบแบบสอบถามเรียงลำดับความสำคัญ (Ranking Question) จากมากไปน้อยหรือน้อยไปมาก ยกตัวอย่างเช่น

กรุณาเรียงลำดับความสำคัญจากมากไปน้อย (1 – 5) ของปัญหาที่พบในการใช้งานระบบปัจจุบัน ....การประมวลผลที่ผิดพลาด ....การรักษาความปลอดภัยของระบบ ....ความยุ่งยากในการนำเข้าข้อมูล ....การประมวลผลที่ใช้ระยะเวลานาน ....รายงานที่ได้ไม่ครบถ้วน

การสร้างคำถามปลายเปิดและคำถามปลายปิดจะมีข้อดีและข้อจำกัดแตกต่างกันไป ดังตารางที่ 3.1

ตารางที่ 3.1 การเปรียบเทียบข้อดีและข้อจำกัดของคำถามแต่ละรูปแบบ

ประเด็น	คำถามปลายเปิด	คำถามปลายปิด
เวลาที่ใช้ในการสร้างคำถาม	ใช้เวลาน้อย	ใช้เวลานาน
เวลาที่ใช้ในการตอบคำถาม	ใช้เวลานาน	ใช้เวลาน้อย
ความครอบคลุมและเจาะลึกในรายละเอียด	มาก	น้อย
การตอบคำถามที่ตรงกับความคิดเห็น	ค่อนข้างสูง	ค่อนข้างต่ำ
การนำข้อมูลไปวิเคราะห์	ค่อนข้างยาก	สะดวกรวดเร็ว

2.4.2 ข้อควรระวังในการใช้แบบคำถาม

1) คำถามควรมีการใช้ข้อความที่อ่านเข้าใจได้ง่าย และมีความชัดเจนว่าต้องการถามถึงอะไร

2) คำถามจะต้องครอบคลุมในเนื้อหาที่ต้องการได้จากผู้ตอบแบบสอบถาม

3) มีการจัดกลุ่มคำถามตามประเด็นหัวข้อต่าง ๆ

4) ควรมีคำอธิบายถึงวัตถุประสงค์ในการจัดทำแบบสอบถาม และวิธีการกรอกแบบสอบถาม

5) ควรมีกำหนดระยะเวลาส่งคืนและวิธีการส่งคืน

6) ควรพิจารณาความจำเป็นในการกรอกข้อมูลส่วนตัวของผู้ตอบแบบสอบถาม ถ้าประเด็นใดไม่ถูกนำมาวิเคราะห์ข้อมูล ก็ไม่จำเป็นต้องให้ผู้ตอบแบบสอบถามเสียเวลาไปโดยเปล่าประโยชน์

7) ในแบบสอบถามควรมีเนื้อที่ส่วนหนึ่งให้ผู้ตอบแบบสอบถามได้แสดงความคิดเห็นข้อเสนอแนะเพิ่มเติม

2.5 การสังเกตการณ์

การสังเกตการณ์ (Observation) เป็นวิธีการที่เป็นที่นิยมเนื่องจากมีคุณค่าในการเสริมสร้างข้อมูลและเพิ่มความเข้าใจที่ลึกซึ้ง ในบริบทที่แบบสัมภาษณ์หรือแบบสอบถามอาจไม่เพียงพอในการแสดงประเด็นหรือลักษณะบางอย่าง การให้สัมภาษณ์หรือการสอบถามมักจะเกี่ยวข้องกับมุมมองและความเข้าใจของบุคคลที่เป็นผู้ถูกสัมภาษณ์ ซึ่งอาจทำให้ข้อมูลที่ได้รับมีความคลุมเครือและความถูกต้องไม่เพียงพอ

ดังนั้น การใช้การสังเกตการณ์เป็นวิธีการสร้างข้อมูลเสริมที่มีคุณค่า ที่ช่วยให้นักวิเคราะห์ระบบเข้าใจประเด็นต่าง ๆ และลักษณะการทำงานในระบบได้อย่างลึกซึ้งกว่าการรับฟังหรือการอ่านข้อความเพียงอย่างเดียว นอกจากนี้ การสังเกตการณ์ยังช่วยให้เน้นไปยังบรรยากาศการทำงาน วัฒนธรรมขององค์กร และปฏิกิริยาในการทำงานของบุคคลที่เกี่ยวข้องกับระบบ ซึ่งสร้างพื้นที่ในการรวบรวมข้อมูลอย่างเป็นธรรมชาติและเชื่อถือได้ นักวิเคราะห์ระบบที่มีประสบการณ์สูงมักจะใช้วิธีการสังเกตการณ์เพื่อนำข้อมูลเข้ามาใช้ในการวิเคราะห์และการวางแผนพัฒนาระบบอย่างมีประสิทธิภาพ โดยสามารถนำความรู้และประสบการณ์เหล่านี้มาปรับปรุงและพัฒนาระบบให้ดียิ่งขึ้นในอนาคต

2.6 การออกแบบแอปพลิเคชันร่วมกัน

การออกแบบแอปพลิเคชันร่วมกัน (Joint Application Design: JAD) เป็นวิธีการที่ได้รับการพัฒนาขึ้นโดยเครือข่ายของผู้ที่เกี่ยวข้องโดยตรงกับการพัฒนาระบบ ซึ่งประกอบด้วยระบบต่าง ๆ ทั้งผู้ใช้ระบบ ผู้บริหารระดับต่าง ๆ ผู้เชี่ยวชาญด้านเทคโนโลยีสารสนเทศ นักวิเคราะห์ระบบ และโปรแกรมเมอร์ หลักการทำงานของ JAD ตั้งอยู่ที่การสร้างการประชุมระดมความคิดของผู้ที่เกี่ยวข้อง โดยมีเป้าหมายหลักคือกำหนดความต้องการการใช้งานระบบร่วมกัน ผู้เข้าร่วมการประชุมสามารถนำเสนอความต้องการในมุมมองของตนเองและแสดงความคิดเห็นที่อาจแตกต่างได้

วิธีการของ JAD ลดระยะเวลาในการรวบรวมข้อมูลความต้องการใช้งานจากผู้ใช้งานระบบ เมื่อเทียบกับวิธีการอื่น ๆ และเปิดโอกาสให้ทุกฝ่ายที่เกี่ยวข้องโดยตรงกับระบบได้มีส่วนร่วมในกระบวนการนำเสนอข้อมูล ข้อมูลที่ได้รับจะรวมถึงความคิดของทุกกลุ่มและจะถูกประมวลผลต่อเนื่องตั้งแต่เริ่มต้นจน ถึงขั้นสุดท้าย ดังนั้นผลที่ได้จากวิธีการนี้จึงมีความน่าเชื่อถือสูงกว่าวิธีการอื่น ๆ

อย่างไรก็ตาม หากไม่มีการจัดการที่ดีในการสร้างความเข้าใจและความร่วมมือกัน วิธีการ JAD อาจส่งผลกระทบต่อโครงการได้เช่นกัน โดยจำเป็นต้องใช้การประชุมและประชุมร่วมกันหลายครั้ง โดยขึ้นอยู่กับขอบเขตของระบบที่ได้กำหนดไว้ในสัญญาว่าจ้าง การที่จะสามารถประสบความสำเร็จตามเป้าหมายจะต้องพึงมีความสามัคคีและความเสียสละอย่างมากจากองค์กร

นอกจากนี้ ผู้นำในการประชุมจำเป็นจะต้องมีความเข้าใจในด้านจิตวิทยาและความสามัคคีที่สูง เพื่อสร้างความกระตือรือร้นและกำลังใจ รวมทั้งสร้างแรงจูงใจให้กับผู้เข้าร่วมประชุมเพื่อระดมสมองและควบคุมให้การระดมสมองแต่ละครั้งเนื่องจากสำคัญในการผลักดันให้ได้ผลลัพธ์ตามที่กำหนดไว้ในแต่ละช่วงเวลา หากขาดคุณลักษณะเหล่านี้ การใช้วิธีการ JAD อาจกลายเป็นการลงทุนที่ไม่คุ้มค่าในที่สุด

2.7 การสร้างโปรแกรมต้นแบบแบบวนซ้ำ

การสร้างโปรแกรมต้นแบบแบบวนซ้ำ (Iterative Prototyping) เป็นเทคนิคที่พัฒนาขึ้นในยุคสมัยปัจจุบัน ซึ่งนักวิเคราะห์ระบบใช้โปรแกรมต้นแบบที่สร้างขึ้นจากความต้องการใช้งานระบบที่ได้รวบรวมไว้ด้วยเป้าหมายในการช่วยให้ผู้ใช้สามารถมองเห็นและทดลองใช้งานระบบใหม่ได้ ซึ่งในกระบวนการนี้มีขั้นตอนการปรับแก้ตามผลสะท้อนกลับจากผู้ใช้และทำซ้ำไปเรื่อย ๆ จนกว่าผู้ใช้จะมีความเห็นร่วมกันว่าโปรแกรมต้นแบบเป็นไปตามความต้องการและเหมาะสม จึงจะเริ่มต้นขั้นตอนต่อไป

คุณค่าของการใช้วิธีการนี้อยู่ที่ผู้ใช้งานสามารถเข้าใจและเกี่ยวข้องกับระบบใหม่ได้มากขึ้นเมื่อเทียบกับวิธีการอื่น ๆ ทำให้การเปลี่ยนแปลงความต้องการในภายหลังมีน้อยลง อย่างไรก็ตาม วิธีการนี้ก็มีข้อจำกัด โดยในกรณีที่ผู้ใช้งานให้ข้อเสนอแนะเพิ่มเติมหรือคำติชมในแต่ละรอบการปรับแก้ มีโอกาสที่กระบวนการจะต้องวนกลับไปแก้ไขจนกระทั่งไม่สามารถหาทางออกที่ดีที่สุดได้ และบางครั้งอาจเกิดกรณีที่โปรแกรมต้นแบบถูกแก้ไขกลับมาสู่รูปแบบเดิมที่เคยถูกปฏิเสธไปแล้วก่อนหน้านี้ เนื่องจากความต้องการและความคิดของผู้ใช้มีแนวโน้มที่เปลี่ยนแปลงตลอดเวลา นอกจากนี้ นักวิเคราะห์ระบบจะต้องมีการคำนึงถึงว่าโปรแกรมต้นแบบถูกสร้างและแก้ไขหลายครั้งอย่างเร่งด่วน เพื่อนำกลับไปให้ผู้ใช้งานยืนยันความต้องการใช้งานอีกครั้ง โครงสร้างของโปรแกรมต้นแบบจึงไม่ได้ผ่านการวิเคราะห์และออกแบบตามหลักการวิศวกรรมซอฟต์แวร์ เพื่อให้เกิดระบบที่สามารถใช้งานได้อย่างมีประสิทธิภาพ

ดังนั้น เทคนิค Iterative Prototyping ได้ถูกพัฒนาขึ้นเพื่อเป็นวิธีการที่เหมาะสมในการพัฒนาระบบที่ให้ความสำคัญกับความเข้าใจและความร่วมมือของผู้ใช้งานในขั้นตอนของการพัฒนาโปรแกรมต้นแบบ แม้ว่าจะมีข้อจำกัดบางอย่าง การนำเข้าใช้เทคนิคนี้ยังคงเป็นทางเลือกที่น่าพิจารณาในการพัฒนาระบบซอฟต์แวร์ในปัจจุบัน

การวิเคราะห์ความต้องการของระบบ

กระบวนการวิเคราะห์ความต้องการของระบบ (System Requirement Analysis) นำผลสรุปความต้องการที่ได้รับมาจากการรวบรวมข้อมูลใช้งานระบบมาเพื่อพิจารณาเอาส่วนที่สามารถนำมาพัฒนาเป็นระบบสารสนเทศ ในขณะเดียวกัน ส่วนที่ไม่เหมาะสมหรือไม่มีความสำคัญเร่งด่วนสามารถกำหนดให้เป็นฟังก์ชันที่จะพัฒนาในเวอร์ชันถัดไปได้ กระบวนการตัดสินใจในการเลือกความต้องการของผู้ใช้นั้นเน้นการรวมทีมและระดมสมองของกลุ่มผู้ใช้ที่มีผู้ที่เกี่ยวข้องอยู่ในระดับทุกระดับ รวมถึงกลุ่มนักพัฒนาระบบด้วย เพื่อให้การตัดสินใจสอดคล้องกับแผนการดำเนินโครงการที่ได้กำหนดไว้ โดยคำนึงถึงทรัพยากรที่มีอยู่สำหรับโครงการ เนื่องจากหากทรัพยากรไม่เพียงพอ จะต้องพิจารณาเลือกความต้องการใช้งานที่มีความสำคัญเร่งด่วนเพื่อนำมาพัฒนาก่อน-หลังตามลำดับ

นอกจากนี้ ณ ปัจจุบันยังมีการวิเคราะห์และปรับปรุงกระบวนการทางธุรกิจให้เข้ากับระบบใหม่ที่จะถูกติดตั้งใช้งานในอนาคต แนวคิดดังกล่าวเรียกว่า "การปรับกระบวนการทางธุรกิจ" (Business Process Redesign : BPR) ซึ่งมีเป้าหมายในการเพิ่มประสิทธิภาพให้กับองค์กรทั้งในด้านการลดต้นทุน และการเพิ่มคุณภาพของผลิตภัณฑ์และบริการ รวมถึงการสร้างความได้เปรียบในการแข่งขันทางธุรกิจ

ดังนั้น กระบวนการวิเคราะห์ความต้องการของระบบเป็นขั้นตอนสำคัญที่เน้นการรวบรวมความเข้าใจและความร่วมมือของผู้ใช้งานในขั้นตอนของการพัฒนาโปรแกรมต้นแบบ ถึงแม้ว่าจะมีข้อจำกัดบางอย่าง แต่การนำไปใช้งานเป็นส่วนหนึ่งของกระบวนการพัฒนาระบบซอฟต์แวร์ในปัจจุบันคงเป็นทางเลือกที่มีความคุ้มค่าในการใช้ในองค์กรและโครงการที่เกี่ยวข้อง

1. ประเภทของความต้องการใช้งานระบบ

ความต้องการของระบบสามารถแบ่งได้เป็น 3 ประเภท คือ ความต้องการที่เป็นฟังก์ชัน ความต้องการที่ไม่เป็นฟังก์ชัน และความต้องการที่เป็นข้อบังคับ ดังนี้

1.1 ความต้องการที่เป็นฟังก์ชัน (Functional Requirement) เป็นการอธิบายถึงหน้าที่หลักของระบบหรือบริการที่ระบบจะต้องมีไว้สำหรับผู้ใช้แต่ละบทบาท ยกตัวอย่างเช่น

1) ระบบจัดการร้านสะดวกซื้อ จะมีการกำหนดความต้องการที่เป็นฟังก์ชันไว้ ดังนี้

1.1) ระบบจะต้องสามารถอ่านค่ารหัสสินค้าจากบาร์โค้ด

1.2) ระบบจะต้องแสดงรายละเอียดสินค้าแต่ละรายการบนหน้าจอการคำนวณค่าสินค้า

1.3) ระบบจะต้องคำนวณเงินค่าสินค้าพร้อมทั้งภาษีมูลค่าเพิ่ม ระบบจะต้องคำนวณเงินทอนพร้อมทั้งจัดพิมพ์ใบเสร็จรับเงิน

1.4) ระบบจะต้องปรับปรุงสินค้าคงเหลือเมื่อได้รับเงินค่าสินค้าแต่ละรายการครบถ้วนแล้ว

1.2 ความต้องการที่ไม่เป็นฟังก์ชัน (Nonfunctional Requirement) เป็นความต้องการ ในส่วนที่ไม่ใช่หน้าที่หรือการให้บริการของระบบ แต่จะเน้นในเรื่องคุณภาพของระบบเป็นหลัก เช่น

1) ประสิทธิภาพของระบบ (Efficiency) คือประสิทธิภาพของเครื่องมือที่ใช้ จำนวนผู้ใช้งานระบบในช่วงเวลาเดียวกัน ปริมาณข้อมูลที่จัดเก็บได้สูงสุด ความยืดหยุ่นของระบบ การสำรอง ข้อมูล และการกู้คืนข้อมูล เป็นต้น

2) ความน่าเชื่อถือ (Reliability) คือความถูกต้อง ความสมบูรณ์ ความเสถียร และความมั่นคงของระบบ เป็นต้น

3) ความสามารถในการใช้ประโยชน์ (Usability) คือ การเข้าถึงข้อมูลภายในระยะเวลาที่กำหนด การสื่อสารที่เหมาะสมและชัดเจนกับผู้ใช้ระบบทุกระดับ และการมีฟังก์ชันที่สนับสนุนการใช้งานระบบ เป็นต้น

4) ความสามารถในการบำรุงรักษา (Maintainability) คือ ความสามารถในการทดสอบระบบและการรองรับการเปลี่ยนแปลงของระบบในอนาคต เป็นต้น

5) ความสามารถในการทำงานในสภาพแวดล้อมที่หลากหลาย (Portability) คือ ความสามารถในการทำงานบนระบบปฏิบัติการที่แตกต่างกัน เป็นต้น

1.3 ความต้องการที่เป็นข้อบังคับหรือเป็นข้อจำกัด (Constraint/Pseudo Requirement) เป็นส่วนหนึ่งของกระบวนการวิเคราะห์ความต้องการของระบบซอฟต์แวร์ ที่มุ่งไปที่การกำหนดขอบเขตและเงื่อนไขที่ต้องเป็นไปตามเพื่อให้ระบบถูกออกแบบให้เข้ากับสิ่งแวดล้อมของระบบนั้น ๆ อย่างเหมาะสม ตัวอย่างเช่น ในกรณีที่มีการกำหนดว่าระบบต้องพัฒนาโดยใช้ซอฟต์แวร์ประเภทเปิดเผยรหัส (Open Source) หรือในกรณีที่ต้องการให้ระบบปฏิบัติตามมาตรฐาน Capability Maturity Model Integration: CMMI ตัวข้อบังคับหรือข้อจำกัดเช่นนี้จะเป็นแนวทางที่ช่วยกำหนดกรอบและข้อกำหนดในการออกแบบและพัฒนาระบบ เพื่อให้สอดคล้องกับความต้องการและเป้าหมายของโครงการ รวมถึงให้เกิดความเป็นไปตามมาตรฐานที่กำหนดไว้ได้อย่างถูกต้องและเหมาะสม

ด้วยเหตุนี้ ข้อบังคับหรือข้อจำกัดนั้นมีบทบาทสำคัญในกระบวนการวิเคราะห์ความต้องการของระบบและช่วยในการเสริมสร้างความเข้าใจร่วมกันระหว่างทีมพัฒนาและผู้รับผิดชอบโครงการ ทั้งนี้เพื่อให้การพัฒนาระบบเป็นไปอย่างมีประสิทธิภาพและสอดคล้องกับความต้องการขององค์กรและผู้ใช้งานในที่สุด

ข้อกำหนดเกี่ยวกับความต้องการของระบบ

เอกสารสำคัญที่ได้มาจากขั้นตอนการวิเคราะห์ความต้องการของระบบเรียกว่า “ข้อกำหนดเกี่ยวกับความต้องการของระบบ” (System Requirement Specification : SRS) ซึ่งเป็นเอกสารที่นักวิเคราะห์ระบบจะต้องสร้างขึ้นเพื่อให้ผู้บริหารและผู้ที่เกี่ยวข้องและมีอำนาจในการตัดสินใจเป็นผู้ลงนามเห็นชอบใน SRS นี้ เอกสารนี้มีจุดประสงค์ในการระบุและบรรยายความต้องการของระบบที่จะถูกพัฒนา รวมถึงรายละเอียดเกี่ยวกับคุณสมบัติการทำงาน และข้อกำหนดอื่น ๆ ที่เกี่ยวข้อง เอกสาร SRS นี้จะเป็นเครื่องมือที่ใช้ในกระบวนการพัฒนาระบบต่อไป โดยจะถูกนำมาใช้เป็นแนวทางในการดำเนินการพัฒนาระบบทุกขั้นตอน จนกระทั่งเข้าสู่ขั้นตอนการตรวจรับระบบในที่สุด

1. คุณลักษณะที่ดีของ SRS

ระบบให้สามารถตอบสนองความต้องการใช้งานขององค์กร ดังนั้น SRS จึงต้องมีคุณลักษณะที่เอื้อประโยชน์ใน SRS เป็นเอกสารสรุปความต้องการของระบบอย่างละเอียด เพื่อนำไปใช้ในการออกแบบและพัฒนาการใช้งาน ดังนี้

1) การแบ่งข้อมูลอย่างชัดเจน คือ ข้อมูลควรกำหนดเป็นข้อ ๆ และจัดเป็นกลุ่มตามความสัมพันธ์กัน เพื่อให้เกิดความง่ายต่อการทำความเข้าใจ และการตรวจสอบความครบถ้วนของความต้องการใช้งานระบบมากกว่าการใช้วิธีการเชิงพรรณนา

2) การระบุเหตุผลและความจำเป็น คือ ใน SRS ต้องระบุถึงเหตุผลและความจำเป็นในการใช้งานฟังก์ชันนั้น ๆ นั่นคือ สิ่งที่จะกำหนดเป็นความต้องการ ของระบบจะต้องมีความจำเป็นสำหรับระบบนั้น ๆ ไม่ว่าจะเป็นความต้องการของกลุ่มผู้ใช้เพียงไม่กี่คน แต่ถ้าจำเป็นต้องใช้งานก็จะต้องระบุไว้ใน SRS ตรงกันข้ามกับสิ่งที่จะมีไว้ในระบบเพียงเพื่อให้ระบบครบถ้วนสมบูรณ์แต่ระบุกลุ่มผู้ใช้งานไม่ได้ ซึ่งกรณีนี้นักวิเคราะห์ระบบไม่ควรระบุไว้ใน SRS

3) ความสามารถในการทดสอบและตรวจสอบผล คือ สิ่งที่จะกำหนดเป็นความต้องการของระบบควรใช้ข้อความที่สามารถนำไปใช้ในการทดสอบระบบ และตรวจสอบผลลัพธ์ที่เกิดขึ้นได้ว่าถูกต้องแม่นยำหรือไม่ เพื่อให้แน่ใจได้ว่าระบบที่พัฒนาขึ้นสามารถตอบสนองความต้องการใช้งานของผู้ใช้ได้จริง

4) ความชัดเจนและเข้าใจง่าย คือ สิ่งที่จะกำหนดเป็นความต้องการของระบบ จะต้องใช้ ข้อความที่ทุกคนในองค์กรสามารถอ่านแล้วเข้าใจได้ถูกต้องตรงกัน โดยเฉพาะอย่างยิ่งการใช้คำศัพท์ทางเทคนิคจะต้องระมัดระวังว่า ผู้อ่านที่ไม่ได้จบการศึกษาทางด้านคอมพิวเตอร์อาจไม่คุ้นเคย และไม่เข้าใจถึงสิ่งที่ได้กล่าวไว้ใน SRS ดังนั้น นักวิเคราะห์ระบบจึงควรใช้คำที่สามารถสื่อสารและ สร้างความเข้าใจร่วมกันของผู้ที่เกี่ยวข้องทุกระดับ

5) ไม่มีข้อขัดแย้ง คือ สิ่งที่จะกำหนดเป็นความต้องการของระบบทั้งหมดจะต้อง ไม่มีข้อใดที่ขัดแย้งกัน เพราะจะนำไปสู่ปัญหาในทางปฏิบัติ ดังนั้น การทบทวน SRS อย่างละเอียดถี่ถ้วนจึงเป็นสิ่งที่สำคัญมาก

6) ความยืดหยุ่นตามสภาพแวดล้อมทางธุรกิจ คือ สิ่งที่จะกำหนดเป็นความต้องการของระบบจะต้องสามารถปรับเปลี่ยนได้ตามความเปลี่ยนแปลงของสภาพแวดล้อมทางธุรกิจ เนื่องจากการพัฒนาระบบค่อนข้างใช้เวลานานพอสมควร ดังนั้น นักวิเคราะห์ระบบควรกำหนด SRS ให้สามารถรองรับการเปลี่ยนแปลงที่อาจเกิดขึ้นระหว่างการดำเนินงานโครงการด้วย

เพื่อสร้างความเข้าใจร่วมกันในการกำหนด SRS ที่สามารถนำไปใช้งานได้จริง จึงขอยกตัวอย่างข้อความที่เหมาะสมสำหรับการจัดทำ SRS ในประเด็นต่าง ๆ ไว้โดยสังเขป ดังตารางที่ 3.2

ตารางที่ 3.2 ตัวอย่างข้อความที่เหมาะสมสำหรับการจัดทำ SRS

ประเด็น	SRS ที่เหมาะสม
การคำนวณเงินที่ลูกค้าต้องจ่ายในการซื้อแต่ละครั้ง	เงินที่ลูกค้าต้องจ่ายจะได้จากผลรวมของยอดเงินค่าสินค้าแต่ละชนิด ซึ่งสินค้าแต่ละครั้งเป็นผลจากการคูณราคาสินค้ากับจำนวนสินค้าชนิดนั้นๆ ที่ลูกค้าสั่งซื้อ
การประมวลผลปริมาณสินค้าที่ต้องสั่งซื้อเข้าสู่คลังสินค้า	ปริมาณสินค้าที่ต้องสั่งซื้อเข้าสู่คลังสินค้าจะได้จากการนำค่าเฉลี่ยของยอดขายสินค้าต่อวันคูณด้วย 20
ประสิทธิภาพของระบบใหม่ที่จะพัฒนา	ระบบใหม่ที่จะพัฒนาจะต้องสามารถเพิ่มยอดขายรวมทั้งปีให้แก่องค์กรไม่น้อยกว่า 10% นับตั้งแต่เริ่มใช้งานระบบจริงเป็นเวลา 1 ปี (12 เดือน)

2. การจัดทำเอกสาร SRS

สำหรับโครงการที่มีขนาดเล็กสามารถจัดทำเอกสาร SRS โดยการสรุปความต้องการของระบบอย่างละเอียดในรูปแบบข้อ ๆ ทำให้เอกสารอ่านง่ายและสามารถแยกประเด็นต่าง ๆ ได้อย่างชัดเจน เพื่อให้ผู้อ่านเข้าใจได้ง่าย และสามารถตรวจสอบความครบถ้วนของความต้องการได้อย่างรวดเร็ว สำหรับโครงการขนาดกลางหรือขนาดใหญ่ ที่มีรายละเอียดความต้องการของระบบจำนวนมาก ควรใช้แบบฟอร์มสำหรับสรุปความต้องการของระบบแต่ละฟังก์ชัน เพื่อให้ได้ข้อมูลเกี่ยวกับแต่ละฟังก์ชันอย่างครบถ้วน และสามารถใช้ในการอ้างอิงและตรวจสอบการทำงานในขั้นตอนการออกแบบและพัฒนาระบบได้อย่างสะดวกและรวดเร็ว

การจัดทำเอกสาร SRS เป็นกระบวนการสำคัญที่ช่วยให้เกิดความเข้าใจและความเห็นร่วมกันเกี่ยวกับความต้องการของระบบ นอกจากนี้ยังช่วยให้ทีมพัฒนาสามารถมีแนวทางในการออกแบบและพัฒนาระบบอย่างเป็นระบบและตรงกับความต้องการของลูกค้าและองค์กร การจัดทำเอกสาร SRS นั้นจึงควรถูกดำเนินการอย่างรอบคอบและตรงไปตามหลักการวิธีการที่เหมาะสมในแต่ละโครงการ สามารถยกตัวอย่างเอกสาร SRS ได้ดังตารางที่ 3.3

ตารางที่ 3.3 ตัวอย่าง SRS ของระบบซื้อขายสินค้าออนไลน์

ชื่อระบบ	ระบบซื้อขายสินค้าออนไลน์
ชื่อฟังก์ชัน	การซื้อขายสินค้าออนไลน์
วัตถุประสงค์	เพื่อให้ผู้ใช้สามารถเลือกและซื้อสินค้าผ่านเว็บไซต์ออนไลน์ได้
วันที่สรุปข้อมูล	1 สิงหาคม 2566
กลุ่มผู้ใช้	ลูกค้า, ผู้ดูแลระบบ, พนักงานจัดส่งสินค้า
ความต้องการของระบบ	- ระบบจะต้องมีรายการสินค้าที่สามารถอัปเดตและแสดงผลให้ลูกค้าเห็นได้ - ลูกค้าต้องสามารถสมัครสมาชิกและเข้าสู่ระบบเพื่อทำการซื้อขายสินค้า - ระบบต้องรองรับการจัดการตะกร้าสินค้า และการชำระเงินออนไลน์ - ระบบจะต้องสร้างใบสั่งซื้อและส่งอีเมลยืนยันการสั่งซื้อถึงลูกค้า - พนักงานจัดส่งสินค้าต้องสามารถตรวจสอบและจัดการสถานะการจัดส่งให้กับลูกค้าได้ - ระบบต้องสามารถรายงานสถิติการซื้อขายและสินค้าที่ค้างคลังได้
เงื่อนไขเพิ่มเติม	- การชำระเงินจะสามารถทำได้ผ่านระบบออนไลน์และระบบการเงินที่เป็นที่ยอมรับ - ระบบจะส่งสินค้าให้กับลูกค้าที่ทำการสั่งซื้อสำเร็จ ภายในระยะเวลาที่กำหนด - การรับคืนสินค้าและการคืนเงินจะใช้เงื่อนไขตามที่ระบุไว้ในเว็บไซต์

สรุปท้ายบทที่ 3

เริ่มต้นด้วยกระบวนการรวบรวมความต้องการใช้งานจากแหล่งข้อมูลต่าง ๆ ที่เกี่ยวข้องกับระบบที่กำลังจะพัฒนา ได้แก่ ผู้ใช้ระบบ ผู้บริหาร ระบบที่มีอยู่ในปัจจุบัน และเอกสารที่เกี่ยวข้อง นักวิเคราะห์ระบบสามารถใช้วิธีการต่าง ๆ เพื่อรวบรวมความต้องการเหล่านี้ เช่น การศึกษาจากเอกสาร การค้นคว้าข้อมูล การสัมภาษณ์ การใช้แบบสอบถาม การสังเกตพฤติกรรม รวมถึงวิธีการสมัยใหม่ คือ การออกแบบแอปพลิเคชันร่วมกัน และการสร้างโปรแกรมต้นแบบแบบวนซ้ำ เพื่อนำข้อมูลมาใช้ในการกำหนดความต้องการของระบบ ซึ่งอาจจะแบ่งออกเป็นฟังก์ชันที่จะพัฒนาตามลำดับความสำคัญ ที่ได้มาจากกระบวนการระดมความคิดร่วมกันของกลุ่มผู้ใช้ทุกระดับและทีมนักพัฒนาระบบ

ขั้นตอนสุดท้ายของการวิเคราะห์ความต้องการระบบ คือ นักวิเคราะห์ระบบจะจัดทำเอกสารสำคัญที่เรียกว่า “ข้อกำหนดความต้องการของระบบ” (SRS) ซึ่งจะระบุความต้องการทั้งที่เป็นฟังก์ชัน ไม่เป็นฟังก์ชัน และข้อบังคับที่เกี่ยวข้องกับระบบ ผู้บริหารและนักพัฒนาระบบจะต้องลงนามเห็นชอบเอกสารนี้ร่วมกัน เพื่อใช้เอกสารเหล่านี้ในกระบวนการออกแบบและพัฒนาระบบจนกระทั่งเข้าสู่ขั้นตอนการตรวจรับระบบ

โดยสรุปแล้ว การวิเคราะห์ความต้องการของผู้ใช้ระบบเป็นกระบวนการที่สำคัญและซับซ้อน เพื่อให้เกิดความเข้าใจและความถูกต้องในการกำหนดความต้องการของระบบที่สอดคล้องกับความต้องการและความพึงพอใจของผู้ใช้ระบบ การใช้วิธีการต่าง ๆ ในการรวบรวมข้อมูลและการใช้งานข้อมูลนี้เป็นสิ่งสำคัญในการสร้างระบบที่มีประสิทธิภาพและเหมาะสมกับการใช้งานจริง

คำถามท้ายบทที่ 3

1. อธิบายวัตถุประสงค์และความสำคัญของการวิเคราะห์ระบบ
2. แหล่งข้อมูลของความต้องการใช้งานระบบออกเป็นกี่ประเภท และแต่ละประเภทมีคุณลักษณะสำคัญเป็นอย่างไร
3. อธิบายรายละเอียดและขั้นตอนในการดำเนินงานของวิธีการรวบรวมความต้องการใช้งานของผู้ใช้ทั้งหมด โดยให้ยกตัวอย่าง 3 วิธีการ
4. อธิบายคุณลักษณะของรูปแบบการรวบรวมความต้องการของผู้ใช้ มาพอสังเขป
5. แบ่งประเภทของความต้องการใช้งานระบบมีกี่ประเภท และเสนอภาพรวมของลักษณะสำคัญในแต่ละประเภท

บทที่ 4
แผนภาพกระแสข้อมูล

แผนภาพกระแสข้อมูล (Data Flow Diagram: DFD) เป็นเครื่องมือที่สำคัญและจำเป็นในกระบวนการวิเคราะห์และออกแบบระบบทั้งในด้านเทคนิคและธุรกิจ เนื่องจากช่วยให้เข้าใจโครงสร้างและกระบวนการทำงานของระบบได้อย่างเป็นระบบและลึกซึ้ง ด้วยการใช้สัญลักษณ์เฉพาะที่ได้รับมาตรฐาน ไม่ว่าจะเป็น สัญลักษณ์ของกระบวนการ กระแสข้อมูล และเก็บข้อมูล ทำให้ความหมายและความสัมพันธ์ระหว่างส่วนประกอบต่าง ๆ ของระบบเป็นไปอย่างชัดเจน หน้าที่หลักของแผนภาพกระแสข้อมูล คือ การแสดงภาพรวมและกระบวนการทำงานของระบบในลักษณะที่เป็นระบบ โดยให้ผู้ที่เกี่ยวข้องเข้าใจได้อย่างง่ายดาย ทำให้เป็นเครื่องมือที่มีประสิทธิภาพในการวิเคราะห์ปัญหาและออกแบบโซลูชันให้เหมาะสมและมีประสิทธิผลสูงขึ้น โดยเราสามารถแสดงกระบวนการต่าง ๆ ในระบบเป็นขั้นตอนลำดับการทำงาน ทำให้เห็นความสัมพันธ์และลำดับขั้นตอนของการทำงานในระบบ การใช้สัญลักษณ์และสัญลักษณ์เชื่อมโยงระหว่างส่วนประกอบทำให้การสื่อสารและการเข้าใจเกี่ยวกับระบบนั้นเป็นไปอย่างเป็นระบบและเข้าใจได้ง่ายยิ่งขึ้น

การใช้แผนภาพกระแสข้อมูล เป็นเครื่องมือที่สำคัญและมีประสิทธิภาพในการช่วยวิเคราะห์และออกแบบระบบให้เหมาะสมและมีประสิทธิผลสูง และช่วยให้ผู้ที่เกี่ยวข้องเข้าใจโครงสร้างและกระบวนการทำงานของระบบได้อย่างชัดเจนและมีความเชื่อมโยงกันอย่างมีประสิทธิภาพ ในบทนี้จะมีหัวข้อหลัก ดังต่อไปนี้

ความรู้เกี่ยวกับแผนภาพกระแสข้อมูล

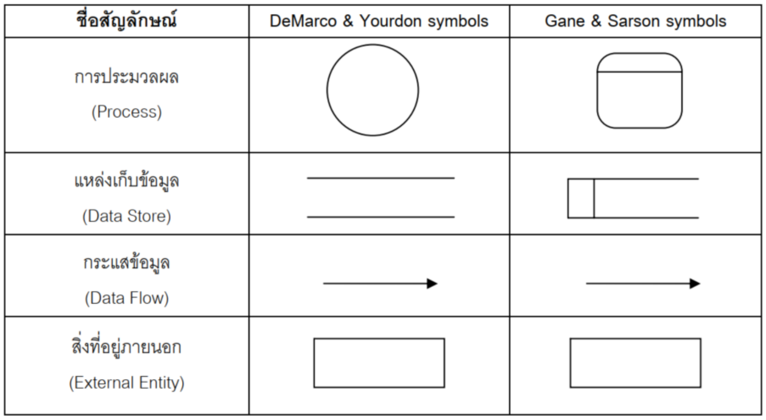
แผนภาพกระแสข้อมูลเป็นแบบจำลองกระบวนการที่นำมาใช้กับการวิเคราะห์และออกแบบระบบเชิงโครงสร้างถูกนำมาใช้ตั้งแต่ยุคที่เริ่มนำภาษาระดับสูงมาใช้โดยแผนภาพดังกล่าวจะแสดงความสัมพันธ์ระหว่างโปรเซสกับข้อมูลเพื่อให้รับทราบว่า ข้อมูลมาจากไหน ข้อมูลไปที่ใด ข้อมูลเก็บไว้ที่ไหน และมีกระบวนการอะไรบ้างที่เกิดขึ้นในระบบ แผนภาพกระแสข้อมูลจะแสดงภาพรวมของระบบและรายละเอียดที่เกี่ยวข้องระหว่างโปรเซสกับข้อมูล แต่นักวิเคราะห์ระบบต้องการทราบรายละเอียดอื่น ๆ ซึ่งก็จะต้องใช้เครื่องมืออื่นเข้าช่วย เช่น คำอธิบายการประมวลผล ตารางการตัดสินใจหรือแบบจำลองข้อมูล ทั้งนี้ขึ้นอยู่กับความต้องการ ใน รายละเอียดเฉพาะเรื่องนั้นๆ เป็นสำคัญ แผนภาพกระแสข้อมูลเป็นแบบจำลองกระบวนการที่ถูกสร้างขึ้นเพื่อวัตถุประสงค์ ดังนี้

1. เป็นแผนภาพเพื่อสรุปภาพรวมของระบบตามแนวทางการวิเคราะห์เชิงโครงสร้าง

2. เป็นข้อตกลงร่วมกันระหว่างนักวิเคราะห์ระบบกับผู้ใช้งาน
3. เป็นแผนภาพที่นำไปใช้ประโยชน์ต่อไปในขั้นตอนการออกแบบระบบ
4. เป็นแผนภาพที่ใช้อ้างอิงหรือเพื่อการปรับปรุง/พัฒนาระบบในอนาคต
5. ทราบที่มาและที่ไปของข้อมูลที่ไหลไปยังกระบวนการต่าง ๆ (Data and Processes)

สัญลักษณ์แผนภาพกระแสข้อมูล

แผนภาพกระแสข้อมูลเป็นวิธีการนำเสนอภาพรวมด้านความต้องการหลักของระบบประกอบด้วย อินพุต เอาต์พุต กระบวนการและข้อมูล โดยทุกคนในทีมงานพัฒนาระบบสามารถมองเห็นระบบได้จากแผนภาพนี้ เพื่อนำไปใช้เป็นแนวทางในการออกแบบระบบสำหรับการสร้างแผนภาพกระแสข้อมูลจะเกี่ยวข้องกับสัญลักษณ์ ทั้ง 5 ด้วยกัน คือ โปรเซส ตาต้าโฟล์วเอ็กซ์เทอร์นัลเอนทิตี ดาตาสโตร์และเรียลไทม์ลิงก์ ดังภาพที่ 4.1 ต่อไปนี้ (โอภาส เอี่ยมสิริวงศ์, 2560)

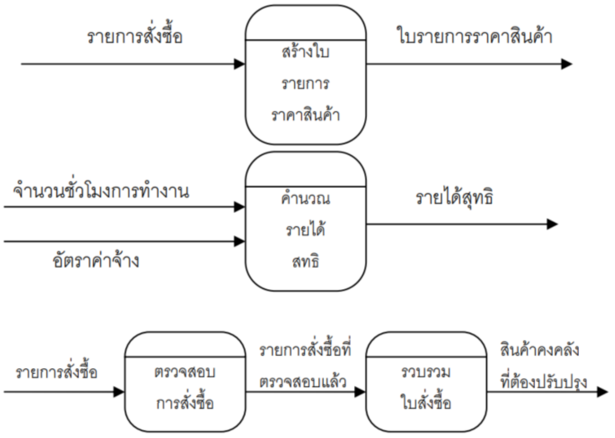


ชื่อสัญลักษณ์	DeMarco & Yourdon symbols	Gane & Sarson symbols
การประมวลผล (Process)		
แหล่งเก็บข้อมูล (Data Store)		
กระแสข้อมูล (Data Flow)		
สิ่งที่อยู่ภายนอก (External Entity)		

ภาพที่ 4.1 สัญลักษณ์แผนภาพกระแสข้อมูล

1. โปรเซส (Processes)

เป็นสัญลักษณ์แทนกิจกรรมที่เกิดขึ้นในระบบหรือกระบวนการที่ต้องทำในระบบ โดยพิจารณาจากภาพที่



ภาพที่ 4.2 ตัวอย่างการใช้สัญลักษณ์การประมวลผล (Process)

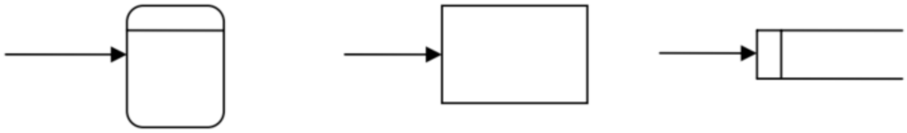
สัญลักษณ์โปรเซสจำเป็นต้องมีหมายเลขกำกับเสมอ ซึ่งเรียกว่า “หมายเลขโปรเซส” โดยมักจะกำหนดเป็นหมายเลข 1, 2, 3 ตามลำดับ แต่การลำดับหมายเลขของโปรเซส มิได้หมายความว่าจะต้องดำเนินกิจกรรมตามเลขลำดับที่กำกับไว้ในโปรเซสแต่อย่างใดและที่สำคัญหมายเลขโปรเซสจะซ้ำกันไม่ได้ ส่วนชื่อที่นำมาใช้กำกับโปรเซสจะใช้คำกริยา ซึ่งหมายถึง การกระทำ เช่น ลงทะเบียนเรียน ชำระเงิน เช่ารถและพิมพ์รายงาน โดยมีจำนวนโปรเซสได้ตั้งแต่ 2 ถึง 7 โปรเซสด้วยกัน เพราะหากมีมากเกินไปกว่านั้นจะทำให้แผนภาพอ่านยากและแลดูซับซ้อนเกินไป แต่อย่างไรก็ตามแม้ว่าจำนวนโปรเซสจะมิได้ถูกระบุไว้เป็นกฎเกณฑ์หรือข้อบังคับแต่อย่างใด แต่ก็มีผู้เชี่ยวชาญได้ให้คำแนะนำว่า จำนวนโปรเซสควรอยู่ในช่วงระหว่าง 7 บวกลบ ด้วย 2 (7 ± 2) หมายความว่าภายในแผนภาพควรมี 5 ถึง 9 โปรเซส ซึ่งถือเป็นจำนวนที่เหมาะสม

โปรเซสในแผนภาพกระแสข้อมูลจะไม่มีการแสดงรายละเอียดเกี่ยวกับวิธีการทำงาน ดังนั้นโปรเซสจึงเปรียบเสมือนกับกล่องดำที่นำเสนอเพียงว่าทำหน้าที่อะไร มีดาตาโฟล์วอะไรที่อินพุตเข้ามา และมีดาตาโฟล์วอะไรที่เอาต์พุตออกไป ส่วนรายละเอียดเกี่ยวกับวิธีการทำงานของแต่ละโปรเซสจะปรากฏอยู่ในแบบจำลองอีกชนิดหนึ่งที่เรียกว่า “คำอธิบายการประมวลผล”

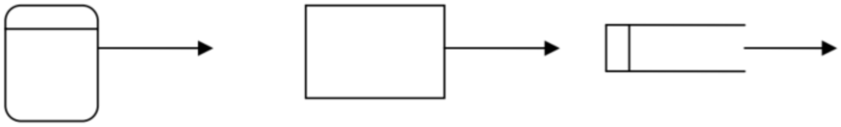
2. ดาตาโฟล์ว (Data Flows)

ดาตาโฟล์ว หรือ กระแสข้อมูล ใช้สัญลักษณ์แทนด้วยเส้นลูกศรที่ไปพร้อมกับข้อมูล กล่าวคือ กระแสข้อมูลเป็นเส้นทางสำหรับให้ข้อมูลเคลื่อนที่ไปยังส่วนที่เกี่ยวข้อง ทำให้เราได้ทราบถึงข้อมูลต่าง ๆ ที่เคลื่อนไหวไปมา

ระหว่างโปรเซสดาตาสโตร์และเอ็กซ์เทอร์นัลเอนทิตีและทุกๆ โปรเซสบนแผนภาพกระแสข้อมูลเมื่อมีดาตาโฟล์วอินพุตเข้าไป ก็จะต้องมีดาตาโฟล์วเอาต์พุตออกมาเสมอ (ไม่ว่าจะเป็นการอินพุตหรือเอาต์พุตมาจากโปรเซสจากดาตาสโตร์หรือจากเอ็กซ์เทอร์นัลเอนทิตีก็ตาม) ดังนั้นโปรเซสที่แต่อินพุตโดยไม่มีเอาต์พุตหรือโปรเซสที่มีแต่เอาต์พุตโดยไม่มีอินพุตใดๆ เข้ามาจึงเป็นสิ่งที่ผิดธรรมชาติ ดังภาพที่ 4.3-4.4



ภาพที่ 4.3 ตัวอย่างสัญลักษณ์กระแสข้อมูลอินพุต

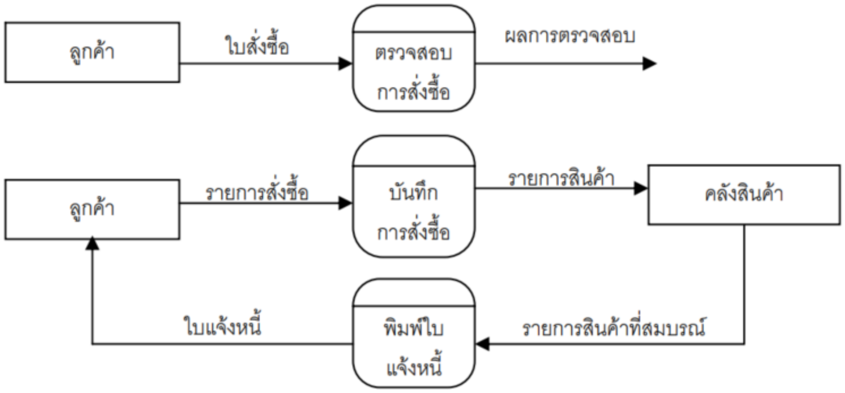


ภาพที่ 4.4 ตัวอย่างสัญลักษณ์กระแสข้อมูลเอาต์พุต

แผนภาพกระแสข้อมูลที่สร้างขึ้นอาจมีดาตาโฟล์วจำนวนมาก ทำให้ดาตาโฟล์วแต่ละเส้นที่ลากโยงไปมาเกิดการตัดกันขึ้นมา ซึ่งขัดแย้งกับหลักการเขียนแผนภาพกระแสข้อมูลที่ดีที่พยายามให้หลีกเลี่ยงเส้นดาตาโฟล์วที่ตัดกัน เพราะจะทำให้แผนภาพแลดูยุ่งเหยิงไม่มีระเบียบ หากจำเป็นต้องมีบางเส้นตัดกันก็ขอให้เกิดน้อยที่สุดเท่าที่จะทำได้และควรปรับเส้นโฟล์วที่ตัดกันให้เป็นเส้นโฟล์วแบบกระโดด (Jump) เพื่อให้เห็นถึงการไหลของเส้นโฟลว์เหล่านั้นมีความชัดเจนมากขึ้น

3. เอ็กซ์เทอร์นัลเอนทิตี (External Entities)

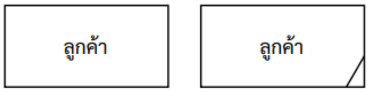
ในแผนภาพกระแสข้อมูลจะมีหน่วยที่อยู่ภายนอกขอบเขต ระบบที่เรียกว่าเอ็กซ์เทอร์นัลเอนทิตี ทำหน้าที่ส่งข้อมูลอินพุตเข้ามายังโปรเซส เพื่อแสดงถึงแหล่งที่มาของข้อมูล (Source) รวมถึงการรับเอาต์พุตจากโปรเซสเพื่อแสดงถึงจุดสิ้นสุดของกระแสข้อมูล (Sink) และด้วยเหตุนี้เอง จึงทำให้สัญลักษณ์นี้มีชื่อเรียกอยู่หลายชื่อด้วยกัน เช่น Source, Sink, Destination, External Agent หรือ Boundary ซึ่งล้วนแต่แทนความหมายเดียวกัน สำหรับสัญลักษณ์นี้จะแทนด้วยรูปสี่เหลี่ยมผืนผ้า ทำหน้าที่รับส่งข้อมูลระหว่างโปรเซสเท่านั้น ไม่สามารถเชื่อมต่อโดยตรงกับดาตาสโตร์หรือระหว่างเอ็กซ์เทอร์นัลเอนทิตีด้วยกัน เนื่องจากไม่สามารถสื่อความหมายใดๆ ได้ ดังภาพที่ 4.5



ภาพที่ 4.5 ตัวอย่างสัญลักษณ์เอ็กซ์เทอร์นัลเอนทิตี

เอ็กซ์เทอร์นัลเอนทิตีสามารถเป็นได้ทั้งบุคคล หน่วยงานหรือระบบงาน ในการพิจารณาหน่วยงานหรือบุคคลที่จะเป็นเอ็กเทอร์นัลเอนทิตีนั้น ใช่ว่าจะนำบุคคลในระบบมากำหนดได้ทั้งหมด ตัวอย่างเช่น ระบบร้านเช่าดีวีดีจะประกอบไปด้วยเอ็กซ์เทอร์นัลเอนทิตีคือ ลูกค้า (Customer) ผู้จัดการ (Manager) และสำนักหักบัญชีบัตรเครดิต (Credit Card Clearinghouse) แต่บางคนอาจออกแบบด้วยการกำหนดให้พนักงาน (Operator) เข้ามาเป็นเอ็กซ์เทอร์นัลเอนทิตีด้วย ซึ่งความจริงนั้นไม่ถูกต้อง เนื่องจากบุคคลใด ๆ ก็ตามที่ปฏิบัติการกับโปรเซสโดยตรง เช่น พนักงานป้อนข้อมูลหรือเสมียนจะถือเป็นส่วนหนึ่งของกระบวนการ ไม่ถือเป็นบุคคลที่อยู่ภายนอกขอบเขตของระบบแต่อย่างใด ซึ่งผู้ปฏิบัติการตามกระบวนการต่าง ๆ อาจมีการอธิบายในรายละเอียดลงในคำอธิบายการประมวลผลข้อมูล แต่จะไม่ปรากฏอยู่ในแผนภาพกระแสข้อมูล ดังนั้นการกำหนดเอ็กซ์เทอร์นัลเอนทิตีจะพิจารณาถึงบุคคลที่ระบบไม่สามารถควบคุมได้เป็นสำคัญ กล่าวคือจะต้องเป็นสิ่งที่อยู่ภายนอกขอบเขตของระบบประมวลผลอย่างไรก็ตาม ก็มีข้อยกเว้นสำหรับบุคคลภายในระบบที่ถือเป็นเอ็กซ์เทอร์นัลเอนทิตีโดยธรรมชาติ เช่น ผู้จัดการ (Managers) หรือทีมงาน (Staff) เป็นต้น

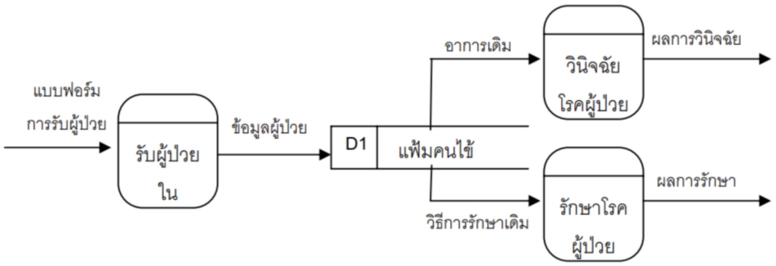
ในการเขียนแผนภาพกระแสข้อมูล เอ็กซ์เทอร์นัลเอนทิตีมักจะถูกจัดวางตำแหน่งให้อยู่บริเวณด้านนอกหรือรอบ ๆ แผนภาพ ทั้งนี้ก็เพื่อให้แผนภาพแลดูสวยงามและง่ายต่อการตรวจสอบ นอกจากนี้สัญลักษณ์ของเอ็กซ์เทอร์นัลเอนทิตียังสามารถทำได้ เนื่องจากข้อจำกัดด้านการเชื่อมโยงของดาตาโฟล์วที่อาจทำให้เส้นโฟล์วดังกล่าวเกิดการตัดกันจึงเป็นสิ่งที่ควรหลีกเลี่ยง เพราะจะทำให้แผนภาพแลดูยุ่งเหยิงและอ่านยาก ดังนั้น การทำซ้ำของเอ็กซ์เทอร์นัลเอนทิตี จึงช่วยขจัดการตัดกันของเส้นโฟล์วได้ โดยจะใช้เครื่องหมาย “\” (Back Slash) กำกับไว้ที่มุมล่างซ้ายของสัญลักษณ์



ภาพที่ 4.6 เอ็กซ์เทอร์นัลเอนทิตีของลูกค้า และสัญลักษณ์การทำซ้ำของลูกค้า

4. ดาตาสโตร์ (Data Stores)

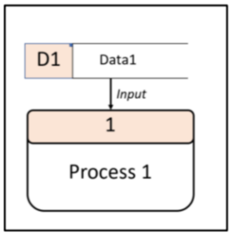
เป็นแหล่งเก็บข้อมูล ซึ่งจะไม่สนใจว่าระบบจะใช้สื่อจัดเก็บข้อมูลประเภทใด ดาตาสโตร์จะต้องมีชื่อข้อมูลที่จัดเก็บพร้อมกับลำดับลาเบลไว้ เช่น D1, D2, D3 ตามลำดับ (อักษร D เป็น คำย่อมาจากคำว่า Data) ดาตาสโตร์เหล่านี้จะถูกใช้งานโดยโปรเซสและสามารถทำได้ ส่วนที่มาของดาตาสโตร์นั้นจะได้มาจากการสร้างแบบจำลองข้อมูล ซึ่งจะกล่าวรายละเอียดในบทที่ 5 การออกแบบฐานข้อมูล สำหรับลูกศรของดาตาโฟล์วที่ใช้เชื่อมโยงระหว่างดาตาสโตร์กับโปรเซสจะมีความหมาย ดังภาพที่ 4.7



ภาพที่ 4.7 ตัวอย่างสัญลักษณ์ดาตาสโตร์ (Data Stores)

4.1) ลูกศรจากดาตาสโตร์ชี้ไปยังโปรเซส (Input)

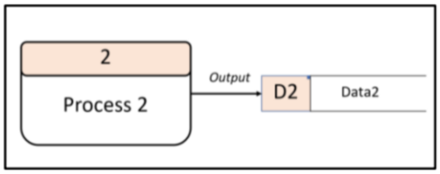
เป็นสัญลักษณ์ของการอินพุต ซึ่งเกี่ยวข้องกับการอ่านหรือการดึงข้อมูลจากดาตาสโตร์ขึ้นมาใช้งาน เช่น อ่านข้อมูลจากแฟ้มลูกค้าหรือดึงรายการสั่งซื้อสินค้าของลูกค้าขึ้นมาใช้งาน เป็นต้น ดังภาพที่ 4.8



ภาพที่ 4.8 แสดงโปรเซสที่มีการอินพุตจากดาตาสโตร์

4.2) ลูกศรจากโปรเซสชี้ไปยังดาตาสโตร์ (Output)

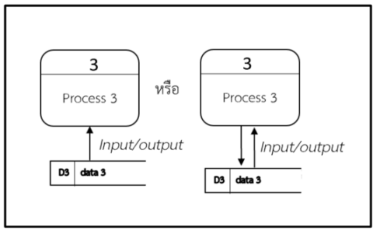
เป็นสัญลักษณ์ของเอาต์พุต ซึ่งเกี่ยวข้องกับการเพิ่มข้อมูลลงในดาตาสโตร์รวมถึงการอัปเดต เช่น การบันทึกข้อมูลนักศึกษารายใหม่หรือการอัปเดตข้อมูลลูกค้า เป็นต้น ดังภาพที่ 4.9



ภาพที่ 4.9 แสดงโปรเซสที่มีการเอาต์พุตเข้าไปยังดาตาสโตร์

4.3) ลูกศรบนปลายทั้งสองด้าน (Input/Output)

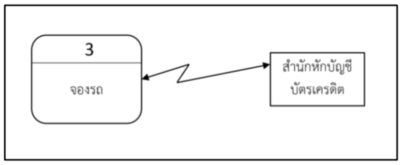
เป็นสัญลักษณ์ของการเป็นได้ทั้งอินพุตและเอาต์พุต ซึ่งเกี่ยวข้องกับการอัปเดตข้อมูลลงใน ดาตาสโตร์ โดยจะดึงข้อมูลจากดาตาสโตร์ขึ้นมาเพื่อปรับปรุง จากนั้นก็จะอัปเดตข้อมูลลงไป เช่น ดึงข้อมูลลูกค้าขึ้นมาเพื่อแก้ไขที่อยู่จากนั้นก็ดำเนินการจัดเก็บลงไปใหม่เพื่อทำการอัปเดต ดังภาพที่ 4.10



ภาพที่ 4.10 แสดงโปรเซสที่มีการจัดการข้อมูลในดาตาสโตร์ทั้งแบบอินพุตและเอาต์พุต

5. เรียลไทม์ลิงก์ (Real-Time Link)

เป็นการเชื่อมโยงสื่อสารระยะไกลที่มีการโต้ตอบกันไปมาระหว่างเอ็กซ์เทอร์นัลเอนทิตีกับโปรเซส โดยจะเป็นการสื่อสารแบบเรียลไทม์ที่มีการโต้ตอบแบบทันทีทันใด ตัวอย่างเช่น การตรวจสอบบัตรเครดิตธนาคารผ่านระบบสื่อสาร เพื่อรอการตอบรับสินเชื่อ จากสำนักหักบัญชีบัตรเครดิต ดังภาพที่ 4.11

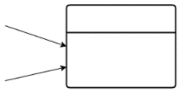
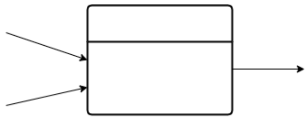
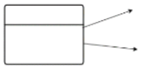
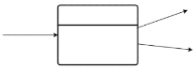
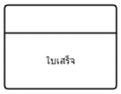
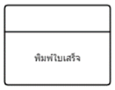
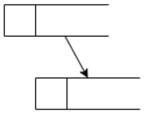
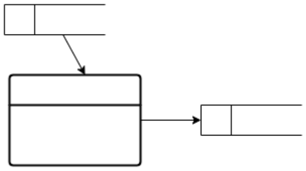
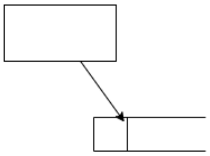
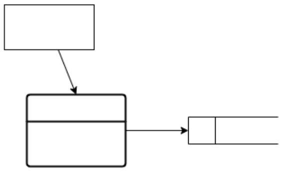


ภาพที่ 4.11 แสดงสัญลักษณ์เรียลไทม์ลิงก์

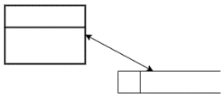
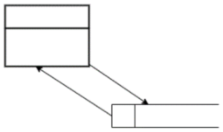
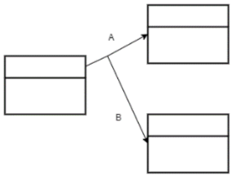
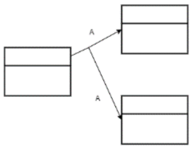
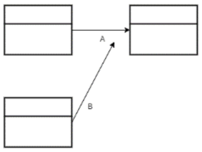
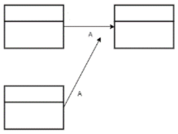
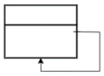
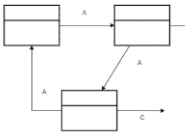
กฎที่ใช้ในการเขียนแผนภาพกระแสข้อมูล

ที่ผ่านมาได้ทราบถึงสัญลักษณ์ต่าง ๆ ที่นำมาสร้างแผนภาพกระแสข้อมูลแล้วแต่ก็ใช่ว่าสัญลักษณ์ทั้งหลายเหล่านั้นจะเชื่อมโยงถึงกันได้ทั้งหมด ดังนั้นจึงควรเรียนรู้ถึงกฎเกณฑ์การเขียนก่อนเพื่อให้ได้มาซึ่งแผนภาพที่ถูกต้องตามหลักการและข้อบังคับ ดังตารางที่ 4.1

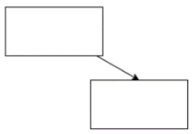
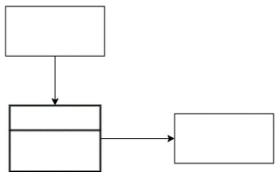
ตารางที่ 4.1 กฎในการสร้างแผนภาพกระแสข้อมูล

ผิด	**ถูก**	**คำอธิบาย**
การประมวลผล (Process)		
		ในการประมวลผลจะมีข้อมูลเข้าอย่างเดียวไม่ได้
		ในการประมวลผลจะมีข้อมูลออกอย่างเดียวไม่ได้
		ข้อความที่บรรจุอยู่ในการประมวลผลจะต้องเป็นคำกริยา หรือคำนามที่แสดงถึงกริยาเท่านั้น
แหล่งเก็บข้อมูล (Data Store)		
		ไม่สามารถส่งข้อมูลโดยตรงจากแหล่งเก็บข้อมูลอันหนึ่งไปยังแหล่งเก็บข้อมูลอีกอันได้โดยตรง ข้อมูลต้องผ่านการจากประมวลผลจึงจะไปที่แหล่งเก็บข้อมูลได้
		ไม่สามารถส่งข้อมูลจากสิ่งที่อยู่ภายนอก (External Entity) ไปยังแหล่งเก็บข้อมูลได้โดยตรง ข้อมูลต้องผ่านจากการประมวลผลที่ได้รับข้อมูลจากแหล่งข้อมูลภายนอกแล้วจึงส่งต่อให้ แหล่งเก็บข้อมูล

ตารางที่ 4.1 กฎในการสร้างแผนภาพกระแสข้อมูล (ต่อ)

ผิด	ถูก	คำอธิบาย
กระแสข้อมูล (Data Flow)		
		กระแสข้อมูล จะมีทิศทางการไหลของข้อมูลเพียงทิศทางเดียวใน 1 กระแสข้อมูล เนื่องจากไม่มีการทำงานใดที่เกิดขึ้นพร้อมกัน
		การแยกข้อมูลออกจาก เส้นกระแสข้อมูลที่ไปสู่การประมวลผลอื่น ๆ พร้อมกันต้องเป็นข้อมูลเดียวกัน
		การเชื่อมข้อมูลจากข้อมูลมากกว่า 1 แหล่งเข้าสู่ เส้นกระแสข้อมูลเดียวกันต้องเป็นข้อมูลเดียวกัน
		กระแสข้อมูลไม่สามารถไหลกลับเข้าสู่การประมวลผลเดิมได้โดยตรง จะต้องมีการประมวลผลอย่างน้อยหนึ่งอย่างจัดการกับข้อมูลดังกล่าวก่อน สร้างกระแสข้อมูลอื่น แล้วจึงคืนค่ากระแสข้อมูลเดิมกลับมายังประมวลผลเริ่มต้น

ตารางที่ 4.1 กฎในการสร้างแผนภาพกระแสข้อมูล (ต่อ)

ผิด	ถูก	คำอธิบาย
สิ่งที่อยู่ภายนอก (External Entity)		
		ไม่สามารถส่งข้อมูลโดยตรงจาก สิ่งที่อยู่ภายนอกระบบอันหนึ่งไปยังสิ่งที่อยู่ภายนอกระบบอันได้โดยตรง ข้อมูลต้องผ่านการจากประมวลผลภายในระบบจึงส่งไปยังสิ่งที่อยู่ภายนอกระบบปลายทางได้

ประเภทของแผนภาพกระแสข้อมูล

การเขียนกระแสข้อมูลนั้นสามารถเขียนได้ 2 แบบ คือแบบตรรกะ (Logical Data Flow Diagram) และแบบกายภาพ (Physical Data Flow Diagram) สามารถอธิบายได้ ดังนี้

1. แผนภาพกระแสข้อมูลแบบตรรกะ (Logical DFD)

แผนภาพนี้จะเป็นการเน้นในส่วนของธุรกิจว่าธุรกิจมีการทำงานอย่างไร มีเหตุการณ์อะไรบ้างที่เกิดขึ้น ข้อมูลที่ต้องการมีอะไรบ้าง และได้ข้อมูลอะไรจากเหตุการณ์นั้น ๆ แต่ไม่ได้บอกว่าระบบจะถูกสร้างอย่างไร

2. แผนภาพกระแสข้อมูลแบบกายภาพ (Physical DFD)

แผนภาพนี้จะบอกได้ว่าจะสร้างระบบอย่างไร ประกอบด้วยฮาร์ดแวร์ ซอฟต์แวร์ ข้อมูล และบุคคลใดที่เกี่ยวข้องกับระบบบ้าง

ตารางที่ 4.2 ข้อแตกต่างระหว่างแผนภาพกระแสข้อมูลแบบตรรกะกับแบบกายภาพ

รูปแบบการออกแบบ	แบบตรรกะ (Logical DFD)	แบบกายภาพ (Physical DFD)
ขั้นตอนการประมวลผล	- แสดงถึงธุรกิจสามารถที่จะดำเนินการได้อย่างไร - เป็นกิจกรรมทางธุรกิจ	- แสดงถึงระบบสามารถที่จะทำงานได้อย่างไร - เป็นโปรแกรม หรือโมดูล หรือขั้นตอนการทำงานด้วยมือ
แหล่งเก็บข้อมูล	- เป็นแหล่งที่เก็บรวบรวมข้อมูลลักษณะใดก็ได้	- เป็น Physical File, Database และ Manual File
ชนิดของแหล่งเก็บข้อมูล	- เป็นการเก็บข้อมูลอย่างถาวร	- เป็นการเก็บข้อมูลที่แยกเป็นไฟล์หลักและแฟ้มรายการ ต้องการประมวลผลในเวลาที่แตกต่างกัน สามารถเชื่อมต่อกันได้โดยใช้แหล่งเก็บข้อมูล
การควบคุมระบบ	- เป็นการควบคุมที่เกี่ยวข้องกับธุรกิจ	- แสดงถึงการควบคุมถึงการป้อนข้อมูลที่ถูกต้อง

ประโยชน์ของการสร้างแผนภาพกระแสข้อมูล

1. ประโยชน์ของการสร้างแผนภาพกระแสข้อมูลแบบตรรกะ (Logical DFD)

1) การสร้างแผนภาพกระแสข้อมูลแบบตรรกะช่วยให้การสื่อสารระหว่างผู้ใช้งานเป็นไปอย่างมีประสิทธิภาพ

2) การใช้แผนภาพกระแสข้อมูลแบบตรรกะช่วยเพิ่มความมั่นคงของระบบ

3) แผนภาพกระแสข้อมูลแบบตรรกะช่วยให้นักวิเคราะห์ระบบเข้าใจการดำเนินงานของระบบได้อย่างชัดเจน

4) การใช้แผนภาพกระแสข้อมูลแบบตรรกะช่วยในการบำรุงรักษาและปรับปรุงระบบให้มีความยืดหยุ่นมากขึ้น

5) แผนภาพกระแสข้อมูลแบบตรรกะช่วยลดความซับซ้อนและง่ายต่อการสร้างแผนภาพกระแสข้อมูลแบบกายภาพต่อไป

2. ประโยชน์ของการสร้างแผนภาพกระแสข้อมูลแบบกายภาพ (Physical DFD) ดังนี้

1) การสร้างแผนภาพกระแสข้อมูลแบบกายภาพเป็นการประมวลผลด้วยระบบมือหรือระบบอัตโนมัติ เพื่อเพิ่มความเร็วและประสิทธิภาพในการประมวลผลข้อมูล

2) แผนภาพแบบกายภาพอธิบายรายละเอียดขั้นตอนการประมวลผลได้อย่างละเอียด เพื่อให้นักวิเคราะห์ระบบและผู้ใช้งานเข้าใจขั้นตอนการทำงานได้อย่างชัดเจน

3) แผนภาพแบบกายภาพแสดงถึงลำดับขั้นตอนการทำงาน เพื่อให้ผู้ใช้งานเห็นภาพรวมของกระบวนการและความสัมพันธ์ระหว่างขั้นตอน ซึ่งช่วยให้การนำเสนอและการติดต่อกับผู้ใช้งานมีประสิทธิภาพ

สรุปท้ายบทที่ 4

แผนภาพกระแสข้อมูลเป็นแบบจำลองกระบวนการประเภทหนึ่งที่ใช้แสดงรายละเอียดของกระแสข้อมูลที่จำเป็นต้องใช้งานในระบบ ดังนั้น ในแผนภาพกระแสข้อมูลจึงประกอบด้วย 4 ส่วนหลักคือ

1. กระบวนการทำงาน (Process)
2. เอ็นทิตี้ภายนอก (External Entity)
3. กระแสข้อมูล (Data Flow)
4. แหล่งจัดเก็บข้อมูล (Data Store)

โดยสามารถแบ่งย่อยแผนภาพจากระบบใหญ่ในภาพรวมทั้งหมดที่เรียกว่า แผนภาพบริบท (Context Diagram) ไปเป็นแผนภาพย่อยระดับต่าง ๆ ได้แก่ แผนภาพกระแสข้อมูลระดับที่ 0 ที่ใช้สำหรับนำเสนอความสัมพันธ์ของกระบวนการทำงานหลักในระบบ หรือแผนภาพกระแสข้อมูลระดับที่ 1 เพื่อแสดงรายละเอียดของกระบวนการทำงานย่อย

ระบบสารสนเทศหนึ่ง ๆ จะแบ่งย่อยระดับของแผนภาพเป็นจำนวนมากน้อยเพียงใดขึ้นอยู่กับความซับซ้อน ของระบบ สิ่งสำคัญคือ นักวิเคราะห์ระบบจะต้องตรวจสอบความสมดุลของแผนภาพว่าแต่ละระดับจะต้องนำเสนอ ข้อมูลที่ครบถ้วนและสอดคล้องกัน

คำถามท้ายบทที่ 4

1. แบบจำลองแบ่งออกเป็นกี่ประเภท และแต่ละประเภทจะมีคุณลักษณะสำคัญอย่างไร
2. อธิบายถึงความสำคัญของแบบจำลองกระบวนการ พร้อมทั้งยกตัวอย่างประกอบ
3. อธิบายความหมายขององค์ประกอบต่าง ๆ พร้อมทั้งแสดงสัญลักษณ์ที่ใช้ในการสร้างแผนภาพกระแสข้อมูล
4. อธิบายถึงกฎสำคัญที่นักวิเคราะห์ระบบต้องคำนึงถึงเมื่อสร้างแผนภาพกระแสข้อมูล
5. อธิบายถึงวิธีการตรวจสอบความสมดุลของแผนภาพกระแสข้อมูลในระดับต่าง ๆ

บทที่ 5
การออกแบบฐานข้อมูล

ในธุรกิจสมัยใหม่ที่เต็มไปด้วยการเชื่อมต่อและการสื่อสารที่รวดเร็ว ฐานข้อมูลเป็นสิ่งที่มีความสำคัญอย่างยิ่งในการออกแบบและพัฒนาระบบ ที่สามารถรองรับความต้องการและเป้าหมายของธุรกิจได้อย่างมีประสิทธิภาพ การออกแบบฐานข้อมูลที่เหมาะสมสามารถช่วยให้ธุรกิจสามารถจัดการข้อมูลอย่างมีประสิทธิภาพ วิเคราะห์ข้อมูลอย่างถูกต้อง และตอบสนองต่อความต้องการของลูกค้าและตลาดได้อย่างมีประสิทธิผล การออกแบบฐานข้อมูลสำหรับธุรกิจสมัยใหม่ควรพิจารณาความต้องการของระบบและธุรกิจ โดยทำการวิเคราะห์และสำรวจความต้องการของข้อมูล รวมถึงการกำหนดโครงสร้างข้อมูลและความสัมพันธ์ระหว่างตาราง (Table Relationship) อย่างรอบคอบ ฐานข้อมูลที่ใช้แบบจำลองแบบสัมพันธ์ (Relational Model) เป็นที่นิยมในการออกแบบฐานข้อมูลเพราะสามารถจัดการข้อมูลที่มีความซับซ้อนได้อย่างมีประสิทธิภาพ การออกแบบฐานข้อมูลที่เหมาะสมสำหรับธุรกิจสมัยใหม่นั้น เน้นไปที่การเก็บรวบรวมข้อมูลอย่างมีระเบียบและเป็นระบบ การกำหนดความสัมพันธ์ระหว่างตารางให้ถูกต้องและสอดคล้องกับความสัมพันธ์ของข้อมูล และการใช้เทคโนโลยีที่เหมาะสมเพื่อเพิ่มประสิทธิภาพในการจัดการข้อมูล เช่น การใช้งานเซิร์ฟเวอร์ฐานข้อมูลที่มีความเร็วสูง การใช้งานระบบเก็บข้อมูลในรูปแบบคลาวด์ (Cloud Database) เพื่อสามารถเข้าถึงข้อมูลได้ทุกที่ทุกเวลา การออกแบบฐานข้อมูลในธุรกิจสมัยใหม่นี้สามารถนำไปสู่ประโยชน์ต่าง ๆ เช่น การวิเคราะห์ข้อมูลเพื่อให้ได้ข้อมูลที่มีค่าความรู้และแนวโน้มที่สามารถใช้ในการวางแผนกลยุทธ์ธุรกิจ การปรับปรุงและพัฒนาผลิตภัณฑ์หรือบริการที่ตอบสนองต่อความต้องการของลูกค้าและการตรวจสอบประสิทธิภาพของธุรกิจ นอกจากนี้ยังมีการใช้ข้อมูลในการตัดสินใจทางธุรกิจที่สอดคล้องกับข้อมูลที่เป็นหลักฐาน ทำให้ธุรกิจสามารถมีการตอบสนองที่ถูกต้องและรวดเร็วต่อการเปลี่ยนแปลงในตลาดและสภาวะเศรษฐกิจได้อย่างเหมาะสม

ในบทนี้จะกล่าวถึงโครงสร้างข้อมูล ปัญหาเกี่ยวกับแฟ้มข้อมูล จากนั้นจะกล่าวถึงแนวทางในการใช้ฐานข้อมูลในการบริหารจัดการข้อมูล องค์ประกอบของฐานข้อมูล รูปแบบของฐานข้อมูล (Data Model) การออกแบบฐานข้อมูลเชิงสัมพันธ์ แบบจำลอง E-R (Entity Relational Modeling) การแปลงแผนภาพ E-R เป็นรีเลชั่น และการนอร์มัลไลเซชั่น และการใช้คำสั่ง SQL

การจัดการข้อมูลแบบดั้งเดิม

1.1 โครงสร้างข้อมูล

โครงสร้างข้อมูลมีรูปแบบเป็นลำดับชั้นโดยเริ่มด้วยหน่วยที่เล็กที่สุดคือ บิต (Bit) ไบต์ (Byte) เขตข้อมูล (Feld) ระเบียนข้อมูล (Record) และไฟล์ (File) ตามลำดับ

บิต (Bit) เป็นหน่วยที่เล็กที่สุดของข้อมูลที่จัดเก็บในระบบคอมพิวเตอร์ ประกอบด้วยเลขฐานสองซึ่งมีสถานะเป็น 0 กับ 1

ไบต์ (Byte) ประกอบด้วยบิดหลาย ๆ บิตมาเรียงต่อกัน เช่น นำ 8 บิตมาเรียงต่อกันเป็น 1 ไบต์ ทำให้สามารถสร้างรหัสแทนข้อมูลเพื่อใช้แทนอักขระ ซึ่งอาจเป็นตัวเลข ตัวอักษร หรือสัญลักษณ์ได้ทั้งหมด 28 ตัว หรือเท่ากับ 256 ตัว

เขตข้อมูล (Field) เป็นการนำข้อมูลหลายอักขระมารวมกันเป็นคำเพื่อให้เกิดความหมาย เช่น ชื่อพนักงานและเงินเดือนพนักงาน เป็นต้น

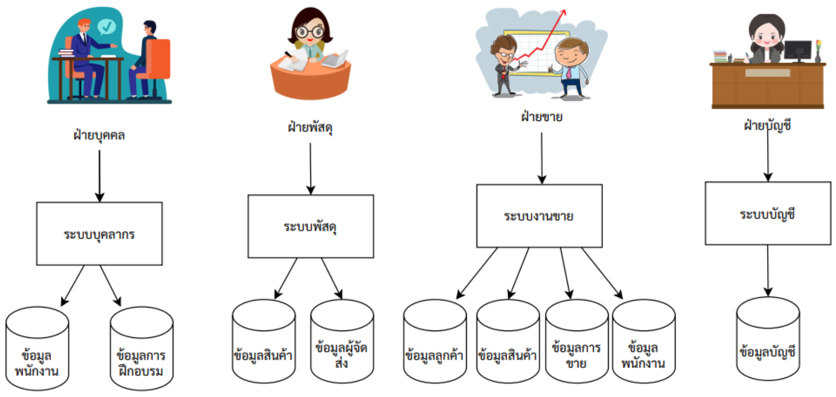
ระเบียนข้อมูล (Record) คือ กลุ่มของเขตข้อมูลที่มีความสัมพันธ์กัน ถูกนำมาไว้รวมกัน เช่น ระเบียนข้อมูลของพนักงาน ประกอบด้วยเขตข้อมูล รหัสพนักงาน ชื่อ-นามสกุล เงินและแผนก เป็นต้น

ไฟล์ (File) คือ กลุ่มของระเบียนข้อมูลที่มีความสัมพันธ์กันถูกนำมาจัดเก็บไว้ด้วยกัน เช่น ไฟล์ประวัติพนักงาน ประกอบด้วย ระเบียนข้อมูลของพนักงานทั้งหมดในองค์การ เป็นต้น

1.2 ปัญหาเกี่ยวกับแฟ้มข้อมูล

องค์การโดยส่วนใหญ่ได้พัฒนาระบบสารสนเทศเพื่อการใช้งานในแต่ละส่วนงาน เช่น ฝ่ายบัญชีได้พัฒนาระบบบัญชีลูกหนี้ ฝ่ายขายได้พัฒนาระบบงานขาย ฝ่ายพัสดุได้พัฒนาระบบพัสดุ และฝ่ายบุคคลได้พัฒนาระบบบุคลากร เป็นต้น แต่ละระบบต่างกันที่มีไฟล์ข้อมูลเป็นของตนเอง และไฟล์ที่สร้างขึ้นอาจเกิดจากการใช้ภาษาโปรแกรมที่แตกต่างกัน เช่น ระบบพัสดุใช้ภาษาซีในการสร้างไฟล์แบบลำดับเชิงดัชนี (Index Sequential File) ในขณะที่ระบบบุคลากรใช้ภาษาจาวาสำหรับสร้างไฟล์แบบลำดับ (Sequential File) การใช้งานแฟ้มข้อมูลมาถึงระดับหนึ่งทำให้เราเข้าใจว่าแต่ละฝ่ายได้จัดทำโปรแกรมและสร้างแฟ้มข้อมูลขึ้นมาใช้งานในบริบทของฝ่ายนั้น ซึ่งส่งผลให้เกิดการจัดเก็บข้อมูลบางส่วนที่ซ้ำซ้อนกัน หากมีการแก้ไขข้อมูลที่เก็บซ้ำซ้อนกันแต่แก้ไขไม่ครบทุกแห่งจะทำให้ข้อมูลไม่สอดคล้องกัน ปัญหาที่เกี่ยวกับแฟ้มข้อมูลสามารถสรุปได้ว่า มีการซ้ำซ้อนของข้อมูล การผูกพัน

ระหว่างข้อมูลและโปรแกรม การไม่สามารถใช้ข้อมูลร่วมกันได้ การขาดความคล่องตัว และการขาดระบบรักษาความปลอดภัยที่ดี (Connolly & Begg, 2014) ดังที่แสดงในภาพที่ 5.1



ภาพที่ 5.1 ระบบแฟ้มข้อมูล

1. การซ้ำซ้อนของข้อมูล (Data Redundancy)

เป็นปัญหาที่เกิดขึ้นเมื่อมีการจัดเก็บข้อมูลที่ซ้ำกันในหลายแห่ง ทำให้ข้อมูลเกิดการซ้ำซ้อนซึ่งไม่จำเป็นและสามารถเกิดความผิดพลาดได้ การซ้ำซ้อนข้อมูลเป็นผลมาจากการเก็บข้อมูลที่ซ้ำกันในแต่ละระบบหรือแฟ้มข้อมูลที่แตกต่างกัน ซึ่งอาจส่งผลให้การแก้ไขข้อมูลเป็นงานที่ยากลำบาก เนื่องจากต้องแก้ไขข้อมูลในทุกแห่งที่เก็บข้อมูลซ้ำซ้อนนั้น การซ้ำซ้อนของข้อมูลทำให้เกิดความไม่สอดคล้องของข้อมูล ความขัดแย้งของข้อมูล และความไม่คล่องตัวของข้อมูลในระบบฐานข้อมูล นอกจากนี้ยังส่งผลให้เกิดการเสียเวลาและทรัพยากรในการจัดการข้อมูลที่ไม่จำเป็น เช่น เนื้อที่จัดเก็บข้อมูลที่เพิ่มขึ้น ค่าใช้จ่ายในการดำเนินการบำรุงรักษาระบบ และความยากลำบากในการควบคุมและรักษาความถูกต้องของข้อมูล

2. ความผูกพันระหว่างข้อมูลและโปรแกรม (Dependence Program-Data)

ความไม่เป็นอิสระของข้อมูล (Data Dependence) เป็นลักษณะที่ข้อมูลและโปรแกรมเชื่อมโยงกันอย่างแนบแน่น โดยมีผลทำให้การเปลี่ยนแปลงโครงสร้างหรือวิธีการใช้งานข้อมูลต้องมีการแก้ไขโปรแกรมที่เกี่ยวข้องด้วย ส่งผลให้เกิดค่าใช้จ่ายสูงในการบำรุงรักษาโปรแกรมเมื่อมีการเปลี่ยนแปลงข้อมูลหรือวิธีการใช้งาน

3. การไม่สามารถใช้ข้อมูลร่วมกันได้ (Data Lack of Sharing)

เกิดจากการจัดเก็บข้อมูลแยกต่างหากโดยแต่ละส่วนงาน ทำให้ข้อมูลไม่สามารถนำมาใช้ร่วมกันได้อย่างมีประสิทธิภาพ และเมื่อมีความต้องการเข้าถึงข้อมูลจากหลายแห่งต้องการการรวมข้อมูลจากหลายแฟ้ม ก็ทำให้ความพร้อมของการใช้งานข้อมูล

4. การขาดความคล่องตัว (Lack of Flexibility)

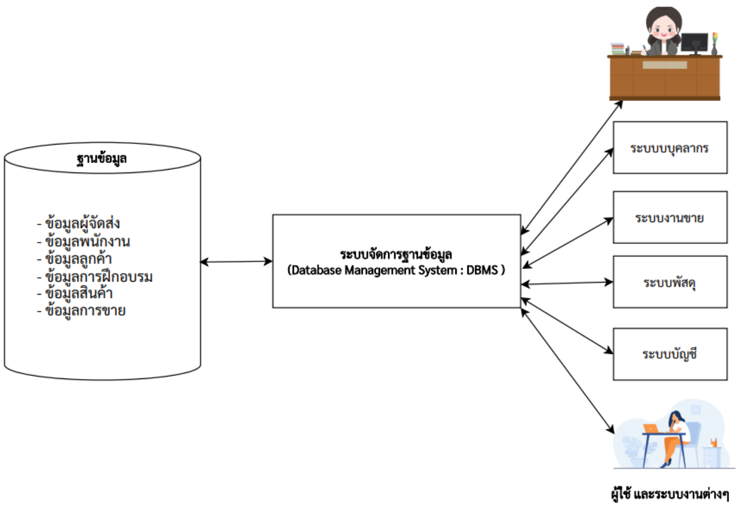
เกิดจากความยากในการปรับเปลี่ยนระบบแฟ้มข้อมูลให้สามารถตอบสนองต่อความต้องการใหม่ โดยที่ระบบแฟ้มข้อมูลไม่สนับสนุนงานในรูปแบบที่ไม่เคยทำเป็นประจำ เช่น การรายงานที่ไม่เคยมีการพิจารณามาก่อน

5. การขาดระบบรักษาความปลอดภัยที่ดี (Poor Security)

เกิดจากข้อจำกัดในการป้องกันการเข้าถึงข้อมูลที่ไม่มีสิทธิ์และการกำหนดสิทธิ์การเข้าถึงข้อมูลที่ไม่ครอบคลุม ส่งผลให้ข้อมูลอาจถูกเข้าถึงโดยผู้ไม่มีสิทธิ์ และการรักษาความปลอดภัยของระบบแฟ้มข้อมูลไม่ครอบคลุมทุกขั้นตอนได้

1.3 แนวทางการใช้ฐานข้อมูลในการบริหารจัดการข้อมูล

เพื่อแก้ไขปัญหาของระบบแฟ้มข้อมูลที่กล่าวมาและเพิ่มประสิทธิภาพในการบริหารจัดการข้อมูล การนำเทคโนโลยีฐานข้อมูลมาใช้ในการบริหารจัดการข้อมูลเป็นแนวทางที่สำคัญ ฐานข้อมูล (Database) เป็นระบบที่ใช้ในการเก็บรวบรวมและจัดเก็บข้อมูลโดยมีโครงสร้างที่เหมาะสมเพื่อให้สามารถเข้าถึงและค้นหาข้อมูลได้อย่างมีประสิทธิภาพ ซึ่งใช้ร่วมกับซอฟต์แวร์ระบบบริหารจัดการฐานข้อมูล (Database Management System: DBMS) เพื่อให้การจัดเก็บและการเรียกใช้ข้อมูลเป็นไปอย่างมีประสิทธิภาพในรูปแบบของโปรแกรมที่ถูกพัฒนามาเป็นเวลานาน การนำข้อมูลมาเก็บไว้ในฐานข้อมูลช่วยลดความซ้ำซ้อนของข้อมูล และเพิ่มความยืดหยุ่นในการเปลี่ยนแปลงและแก้ไขข้อมูล แสดงดังภาพที่ 5.2 โดยสรุปประโยชน์ของระบบจัดการฐานข้อมูล มีดังนี้ (Connolly & Begg, 2014)



ภาพที่ 5.2 ระบบจัดการฐานข้อมูล

1. ลดความซ้ำซ้อนของข้อมูลการ

ลดความซ้ำซ้อนของข้อมูลเป็นส่วนสำคัญในการบริหารจัดการข้อมูลอย่างมีประสิทธิภาพ โดยการใช้ฐานข้อมูลที่เหมาะสมเป็นแนวทางที่สำคัญในการลดความซ้ำซ้อนของข้อมูล ระบบฐานข้อมูลที่ใช้ฐานข้อมูลจัดเก็บข้อมูลอย่างเป็นระบบและมีระเบียบการทำงานที่ชัดเจน สามารถช่วยลดความซ้ำซ้อนของข้อมูลได้ โดยใช้เทคโนโลยีที่เหมาะสม เช่น การใช้คีย์หลัก (primary key) เพื่อระบุแถวข้อมูลที่ไม่ซ้ำกัน การใช้คีย์นอก (foreign key) เพื่อกำหนดความสัมพันธ์ระหว่างตาราง และการใช้ข้อกำหนดความสัมพันธ์ (integrity constraints) เพื่อรักษาความถูกต้องและความสอดคล้องของข้อมูล นอกจากนี้ยังสามารถใช้การนำเข้าข้อมูลเป็นชุดเดียวกันเพื่อลดความซ้ำซ้อน และการใช้เทคนิคการรวมข้อมูลเพื่อตัดหรือลดส่วนที่ซ้ำกันออกไป ซึ่งทั้งหมดนี้ช่วยให้ลดความซ้ำซ้อนของข้อมูลและเพิ่มประสิทธิภาพในการจัดการข้อมูล

2. มีความเป็นอิสระของข้อมูล

ในระบบฐานข้อมูลที่มีความเป็นอิสระของข้อมูล ข้อมูลถูกเก็บรวบรวมในแหล่งรวมข้อมูลที่เป็นส่วนกลาง (Repository) และระบบจัดการฐานข้อมูล (DBMS) ทำหน้าที่ในการรับผิดชอบในการปรับเปลี่ยนโครงสร้างข้อมูล นั่นหมายความว่า โปรแกรมต่าง ๆ ที่เข้าถึงฐานข้อมูลนั้นเป็นอิสระและไม่ได้ผูกพันกับโครงสร้างข้อมูล เมื่อโครงสร้างข้อมูลเปลี่ยนแปลง โปรแกรมยังคงทำงานได้อย่างถูกต้องโดยไม่ต้องปรับเปลี่ยน ซึ่งมีผลให้การพัฒนาและการบำรุงรักษาระบบงานเป็นไปอย่างมีความยืดหยุ่นและมีประสิทธิภาพสูงขึ้น

3. สนับสนุนการใช้ข้อมูลร่วมกัน

การจัดเก็บข้อมูลในส่วนกลางช่วยสนับสนุนให้ผู้ใช้งานภายในองค์กรสามารถใช้ข้อมูลร่วมกันได้ โดยโปรแกรมประยุกต์ใหม่สามารถนำข้อมูลที่มีอยู่แล้วมาใช้ได้โดยไม่ต้องเพิ่มข้อมูลใหม่เข้าสู่ระบบอีกครั้ง

4. มีความคล่องตัวในการใช้งาน

การรวบรวมข้อมูลไว้ในส่วนกลางและการใช้งานผ่านฐานข้อมูลช่วยเพิ่มความคล่องตัวในการใช้งาน เนื่องจากโปรแกรมสามารถเข้าถึงและนำข้อมูลที่อยู่ในระบบฐานข้อมูลมาใช้ได้โดยไม่จำเป็นต้องออกแบบข้อมูลใหม่ นอกจากนี้ ฐานข้อมูลยังมีเครื่องมือที่สนับสนุนการสร้างแบบฟอร์มรายงานต่าง ๆ ที่ช่วยลดขั้นตอนและเวลาในการจัดทำรายงานและการเขียนโปรแกรมอีกด้วย

5. มีระบบรักษาความปลอดภัยของข้อมูลสูง

การรักษาความปลอดภัยของข้อมูลในระบบฐานข้อมูลเป็นสิ่งสำคัญ โดย DBMS จะมีการตรวจสอบและควบคุมการเข้าถึงข้อมูล โดยผู้ใช้จะต้องผ่านระบบการรับรหัสเข้าสู่ระบบและจะได้รับสิทธิ์เฉพาะตามที่กำหนดไว้ ทำให้ผู้ใช้สามารถเข้าถึงและจัดการข้อมูลได้ตามสิทธิ์ที่กำหนด ระบบฐานข้อมูลยังมีการป้องกันการเข้าถึงข้อมูลโดยตรงผ่านทาง DBMS เพื่อป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต

องค์ประกอบของระบบฐานข้อมูล

ระบบฐานข้อมูลประกอบด้วยองค์ประกอบหลัก 4 ส่วนที่มีบทบาทสำคัญในการเก็บรักษาและจัดการข้อมูล ดังนี้

1. ข้อมูล (Data) องค์ประกอบหลักของระบบฐานข้อมูลคือข้อมูลที่จัดเก็บอยู่ในระบบ ซึ่งสามารถเป็นข้อมูลในรูปแบบต่าง ๆ เช่น ข้อความ ตัวเลข รูปภาพ หรือไฟล์เอกสาร ข้อมูลนี้ถูกจัดเก็บในรูปแบบที่เหมาะสมตามโครงสร้างข้อมูลที่กำหนดในระบบฐานข้อมูล

2. ฮาร์ดแวร์ (Hardware) องค์ประกอบที่เกี่ยวข้องกับฮาร์ดแวร์เป็นอุปกรณ์ที่ใช้ในการเก็บข้อมูล รวมถึงเครื่องคอมพิวเตอร์ ฮาร์ดดิสก์ หน่วยความจำ และอุปกรณ์รอบข้างที่เกี่ยวข้อง ฮาร์ดแวร์เป็นส่วนที่จำเป็นในการเก็บรักษาและประมวลผลข้อมูลในระบบฐานข้อมูล

3. ซอฟต์แวร์ (Software) องค์ประกอบซอฟต์แวร์ในระบบฐานข้อมูลมีบทบาทสำคัญในการจัดการข้อมูล ซอฟต์แวร์ระบบฐานข้อมูลประกอบด้วยระบบปฏิบัติการ (Operating Systems) ซึ่งเป็นพื้นฐานในการทำงานของระบบฐานข้อมูล และระบบจัดการข้อมูล (Database Management System: DBMS) ที่มีหน้าที่รับผิดชอบใน

การจัดการข้อมูลในฐานข้อมูล โดย DBMS จะให้ความสามารถในการเก็บ ค้นหา ประมวลผล และควบคุมข้อมูล รวมถึงการจัดการความปลอดภัยและการเข้าถึงข้อมูลในระบบฐานข้อมูล

4. ผู้ใช้ (Users) ผู้ใช้เป็นองค์ประกอบที่สำคัญในระบบฐานข้อมูล เนื่องจากเป็นผู้ใช้งานที่ต้องการใช้ข้อมูลและระบบฐานข้อมูลเพื่อรองรับงานหรือกิจกรรมขององค์การ ผู้ใช้ภายในองค์การอาจเป็นผู้บริหารฐานข้อมูล (Database Administrator: DBA) ผู้เขียนโปรแกรม (Programmers) หรือผู้ใช้งานทั่วไปที่ใช้ข้อมูลจากระบบฐานข้อมูลในการดำเนินงานประจำวัน

แบบจำลองข้อมูล (Data Model)

โครงสร้างและรูปแบบที่ใช้ในการแสดงและอธิบายข้อมูลในระบบฐานข้อมูล โดยที่เน้นการกำหนดความสัมพันธ์และความเกี่ยวข้องระหว่างข้อมูลต่าง ๆ ในระบบ นอกจากนี้ยังให้การกำหนดคุณสมบัติและข้อจำกัดที่เกี่ยวข้องกับข้อมูล ซึ่งช่วยให้ผู้ใช้งานสามารถเข้าใจและใช้งานข้อมูลได้อย่างเหมาะสม แบบจำลองข้อมูลจะช่วยให้ผู้ออกแบบระบบฐานข้อมูลสามารถกำหนดโครงสร้างข้อมูล และเขียนคำสั่งคิวรีเพื่อใช้งานข้อมูลในระบบได้อย่างมีประสิทธิภาพและถูกต้อง (Connolly, T. M., & Begg, C. E., 2014) (Elmasri, R., & Navathe, S. B., 2015)

1. ประเภทของแบบจำลอง

แบบจำลองข้อมูลสามารถแบ่งออกเป็นสองประเภทหลัก คือ แบบจำลองเชิงแนวคิด (Conceptual Data Models) และแบบจำลองเพื่อการนำไปใช้ (Implement Data Models) ดังต่อไปนี้

1.1 แบบจำลองเชิงแนวคิด (Conceptual Data Models)

แบบจำลองเชิงแนวคิดเป็นแบบจำลองที่ใช้ในการออกแบบและวางโครงสร้างของระบบฐานข้อมูล โดยเน้นที่ความสัมพันธ์และความเกี่ยวข้องระหว่างข้อมูล แทนที่จะให้ความสำคัญกับรายละเอียดของการเก็บรักษาข้อมูลและการเข้าถึงข้อมูล การออกแบบแบบจำลองเชิงแนวคิดใช้เครื่องมือแบบสัญลักษณ์เพื่อแสดงข้อมูลและความสัมพันธ์ระหว่างข้อมูล ตัวอย่างของแบบจำลองเชิงแนวคิด ได้แก่ แบบจำลองแบบเอ็นทิตี้ (Entity-Relationship Model) ที่ใช้สัญลักษณ์ เช่น องค์ประกอบหลัก องค์ประกอบลูก และความสัมพันธ์ระหว่างองค์ประกอบในการอธิบายและแสดงความสัมพันธ์ระหว่างข้อมูล

1.2 แบบจำลองเพื่อการนำไปใช้ (Implement Data Models)

แบบจำลองเพื่อการนำไปใช้เป็นแบบจำลองที่ใช้ในกระบวนการนำแบบจำลองเชิงแนวคิดมาประยุกต์ใช้ในระบบฐานข้อมูลจริง โดยใช้เทคนิคและเครื่องมือในการแปลงแบบจำลองเชิงแนวคิดเป็นโครงสร้างฐานข้อมูลที่สามารถนำไปใช้งานได้จริง ตัวอย่างของแบบจำลองเพื่อการนำไปใช้ เช่น แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model) ที่กำหนดโครงสร้างของตารางและความสัมพันธ์ระหว่างตารางในระบบฐานข้อมูลที่ใช้งานจริง และแบบจำลองฐานข้อมูลเชิงวัตถุ (Object-Oriented Database Model) ที่ใช้ในการแสดงและบริหารจัดการวัตถุและความสัมพันธ์ของวัตถุในระบบฐานข้อมูล

2. แบบจำลองฐานข้อมูล (Database Model)

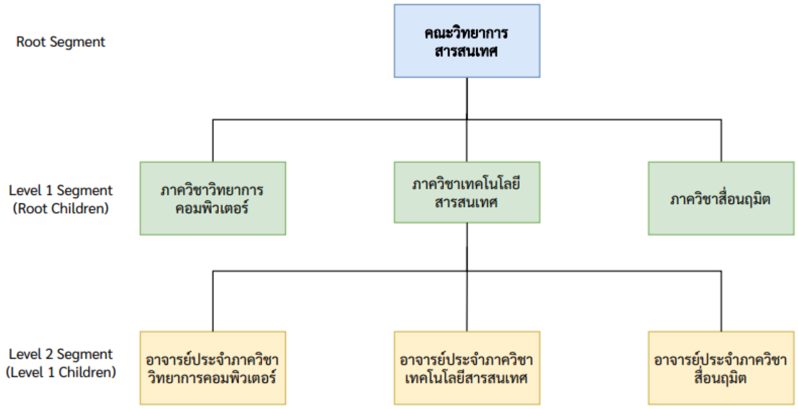
เป็นการกำหนดโครงสร้างและรูปแบบของฐานข้อมูลที่ใช้ในการเก็บรักษาและจัดการข้อมูลในระบบฐานข้อมูล แบบจำลองฐานข้อมูลช่วยให้ผู้ออกแบบระบบฐานข้อมูลเข้าใจและเรียนรู้วิธีการเก็บรักษาข้อมูลในระบบรวมถึงวิธีการเข้าถึงและจัดการข้อมูลในฐานข้อมูล แบบจำลองฐานข้อมูลมี 5 รูปแบบ ดังต่อไปนี้

2.1 แบบจำลองฐานข้อมูลลำดับชั้น (Hierarchical Database Model)

แบบจำลองฐานข้อมูลลำดับชั้น เป็นแบบจำลองที่ใช้โครงสร้างข้อมูลแบบลำดับชั้น เพื่อจัดเก็บและออกแบบระบบฐานข้อมูล โดยมีลักษณะเป็นต้นไม้ที่ประกอบด้วยโหนดหลายชั้นที่เชื่อมต่อกัน โหนดบนสุดของต้นไม้เรียกว่าโหนดหลักหรือโหนดราก (Root Node) และโหนดลูกเป็นโหนดที่เชื่อมต่อกับโหนดบนตัวเอง แบบจำลองฐานข้อมูลลำดับชั้นจะมีโครงสร้างเรียงลำดับเป็นเซกเมนต์ (Segment) ซึ่งสามารถพิจารณาเป็นเหมือนรายการข้อมูลในระบบแฟ้มข้อมูล แบบจำลองฐานข้อมูลรูปแบบนี้เหมาะสำหรับการจัดเก็บข้อมูลที่มีความสัมพันธ์แบบหนึ่งต่อหลาย (one-to-many relationship) โดยที่โหนดหลักสามารถแตกสาขาออกเป็นโหนดลูกได้หลายโหนด ในขณะที่โหนดลูกจะมีโหนดพ่อเพียงตัวเดียว แสดงดังภาพที่ 5.3 สำหรับข้อดีและข้อจำกัดของแบบจำลองรูปแบบนี้ มีดังนี้

ข้อดีของแบบจำลองฐานข้อมูลลำดับชั้น คือ โครงสร้างข้อมูลที่มีความเรียบง่าย เนื่องจากโครงสร้างลำดับชั้นเป็นโครงสร้างข้อมูลที่มีความเรียบง่ายและชัดเจน โดยมีโหนดหลักเป็นต้นไม้และโหนดลูกเป็นสาขาหรือรายละเอียดเพิ่มเติม การจัดเก็บข้อมูลในลำดับชั้นช่วยให้การเข้าถึงและการค้นหาข้อมูลเป็นไปได้อย่างมีประสิทธิภาพ

ข้อจำกัดของแบบจำลองฐานข้อมูลลำดับชั้น คือ ความจำกัดในการแสดงความสัมพันธ์ แบบจำลองฐานข้อมูลลำดับชั้นไม่รองรับความสัมพันธ์แบบหลายต่อหลาย (many-to-many relationship) ที่จำเป็นต่อการจัดเก็บข้อมูลบางรายการ ในกรณีที่ต้องการความสัมพันธ์ที่ซับซ้อนขึ้น



ภาพที่ 5.3 แบบจำลองฐานข้อมูลลำดับชั้น

2.2 แบบจำลองฐานข้อมูลเครือข่าย (Network Database Model)

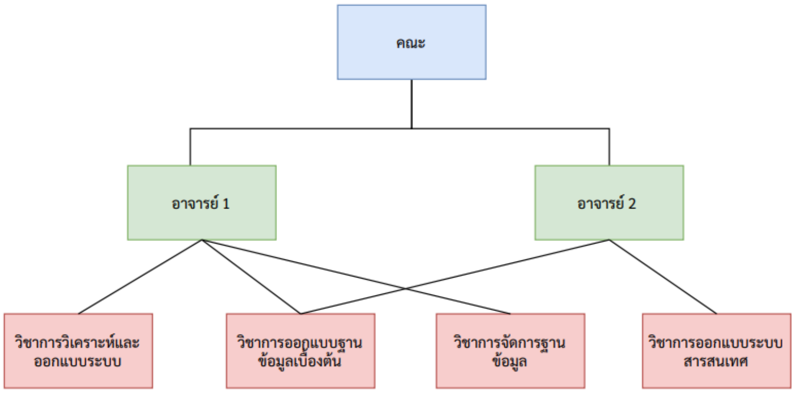
แบบจำลองฐานข้อมูลเครือข่ายเป็นแบบจำลองที่พัฒนาต่อจากแบบจำลองฐานข้อมูลลำดับชั้น แต่มีความสามารถในการแสดงความสัมพันธ์ที่ซับซ้อนและข้ามลำดับชั้นได้ โดยใช้โครงสร้างแบบกราฟที่เชื่อมโยงข้อมูลด้วยการใช้ลิงก์ (Link) ซึ่งให้ความยืดหยุ่นในการแสดงความสัมพันธ์ข้ามลำดับชั้นระหว่างข้อมูล แบบจำลองนี้เหมาะสำหรับรูปแบบข้อมูลที่มีความซับซ้อนและความสัมพันธ์ที่หลากหลาย แต่การใช้งานและการดูแลรักษาในระดับที่สูงอาจเป็นไปได้ยากขึ้น แสดงดังภาพที่ 5.4 สำหรับข้อดีและข้อจำกัดของแบบจำลองรูปแบบนี้ มีดังนี้

1) ข้อดี

ข้อดีของแบบจำลองฐานข้อมูลเครือข่าย คือ ความสามารถแสดงความสัมพันธ์ที่ซับซ้อนระหว่างข้อมูลในระบบฐานข้อมูลได้ โดยการใช้ลิงก์ในการเชื่อมโยงข้อมูลระหว่างเซกเมนต์ ทำให้สามารถแสดงความสัมพันธ์แบบหลายต่อหลาย (many-to-many relationship) รวมถึงความสัมพันธ์ที่ข้ามลำดับชั้นได้

2) ข้อจำกัด

ข้อจำกัดของแบบจำลองฐานข้อมูลเครือข่าย คือ ความซับซ้อนในการใช้งานและการดูแลรักษา เนื่องจากการออกแบบและการใช้งานฐานข้อมูลเครือข่ายมีความซับซ้อนมากกว่าแบบฐานข้อมูลลำดับชั้น การสร้างและการแก้ไขโครงสร้างข้อมูลต้องมีการใช้ลิงก์อย่างถูกต้องและพิจารณาให้รอบคอบ นอกจากนี้ การดูแลรักษาและการจัดการฐานข้อมูลเครือข่ายใช้เทคนิคที่ซับซ้อนมากกว่าแบบฐานข้อมูลอื่นๆ



ภาพที่ 5.4 แบบจำลองฐานข้อมูลเครือข่าย

2.3 แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)

แบบจำลองฐานข้อมูลเชิงสัมพันธ์เป็นแบบจำลองที่ใช้โครงสร้างและการเก็บรักษาข้อมูลในรูปแบบของตาราง (Table) โดยข้อมูลถูกจัดเก็บในตารางที่ประกอบด้วยแถว (Row) และคอลัมน์ (Column) โดยแต่ละแถวในตารางจะแสดงข้อมูลของสิ่งที่เกี่ยวข้องกัน และแต่ละคอลัมน์จะแสดงข้อมูลในลักษณะที่เป็นคุณลักษณะของข้อมูล แบบจำลองนี้เป็นแบบจำลองที่ได้รับความนิยมและการนำไปใช้งานอย่างแพร่หลาย มีความยืดหยุ่นในการออกแบบและการเข้าถึงข้อมูล สามารถทำการสืบค้น แก้ไข และลบข้อมูลได้ง่าย และเป็นรูปแบบที่มีประสิทธิภาพในการจัดการข้อมูลในระบบฐานข้อมูลขนาดใหญ่ แสดงดังภาพที่ 5.5 สำหรับข้อดีและข้อจำกัดของแบบจำลองรูปแบบนี้ มีดังนี้

1) ข้อดี

ข้อดีของแบบจำลองฐานข้อมูลเชิงสัมพันธ์ คือ แบบจำลองฐานข้อมูลเชิงสัมพันธ์ใช้เทคนิคตรวจสอบที่ชัดเจนในการออกแบบฐานข้อมูล โดยใช้ระบบความสัมพันธ์และกฎของแฟ้มข้อมูล ทำให้มีความน่าเชื่อถือและความมั่นใจในความถูกต้องของข้อมูลที่จัดเก็บในฐานข้อมูล อีกทั้ง ระบบตารางในแบบจำลองฐานข้อมูลเชิงสัมพันธ์ช่วยให้สามารถเข้าถึงข้อมูลได้ง่าย โดยใช้ภาษาสอบถามข้อมูลที่เป็นรูปแบบมาตรฐาน เช่น SQL (Structured Query Language) ทำให้ผู้ใช้งานสามารถดึงข้อมูลที่ต้องการได้อย่างรวดเร็วและง่ายดาย

2) ข้อจำกัด

ข้อจำกัดของแบบจำลองฐานข้อมูลเชิงสัมพันธ์ คือ ออกแบบและกำหนดความสัมพันธ์ในตารางและการกำหนดกฎของแฟ้มข้อมูลอาจเป็นไปได้ยากในระบบที่มีความซับซ้อนหรือมีความสัมพันธ์ที่ซับซ้อนมาก และในบางกรณีที่มีการสืบค้นข้อมูลที่ซับซ้อนและมีตารางที่มีขนาดใหญ่ การทำงานกับฐานข้อมูลเชิงสัมพันธ์อาจใช้เวลานานขึ้นและมีประสิทธิภาพที่ต่ำลง

แอตทริบิวต์ (Attributes)

ทูเพิล (Tuples)

รหัสพนักงาน	ชื่อพนักงาน	แผนก	ตำแหน่ง
001	นายพิชิต รัตนา	ไอที	พนักงานพัฒนาระบบ
002	นายเจริญสิทธิ์ มีชัย	บัญชี	พนักงานบัญชี
003	นางสาวรัตนา สดใส	บุคคล	ผู้จัดการฝ่ายบุคคล

ภาพที่ 5.5 แบบจำลองฐานข้อมูลเชิงสัมพันธ์

2.4 แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-Oriented Database Model)

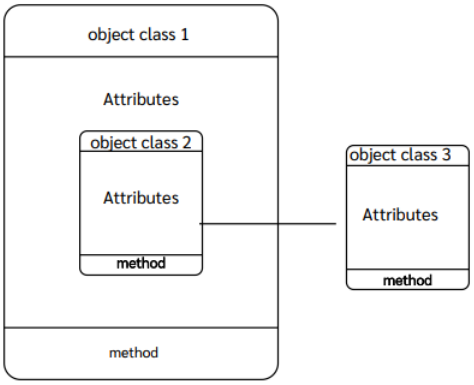
แบบจำลองฐานข้อมูลเชิงวัตถุเป็นแบบจำลองที่ใช้สำหรับการจัดการข้อมูลแบบวัตถุ ซึ่งเกิดจากแนวคิดการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming : OOP) โดยใช้คอนเซปต์ของวัตถุ (Object) และคลาส (Class) ในการเก็บรักษาข้อมูล แบบจำลองนี้สามารถสร้างความสัมพันธ์ระหว่างวัตถุ และสามารถใช้งานฟังก์ชันและความสามารถของวัตถุเพื่อจัดการข้อมูลได้อย่างยืดหยุ่น แบบจำลองนี้เหมาะสำหรับระบบที่มีความซับซ้อนและต้องการการจัดการข้อมูลที่ยืดหยุ่นและก้าวหน้า เช่น ระบบการจัดการภาพ ระบบสารสนเทศทางเทคนิค และระบบสารสนเทศธุรกิจ นอกจากนี้ยังสนับสนุนคุณสมบัติของการนำกลับมาใช้ใหม่ แบบจำลองฐานข้อมูลเชิงวัตถุ จึงถือเป็นเทคโนโลยีใหม่ของ DBMS ที่มักนำไปใช้กับหน่วยงานขนาดใหญ่ แสดงดังภาพ 5.6 สำหรับข้อดีและข้อจำกัดของแบบจำลองรูปแบบนี้ มีดังนี้

1) ข้อดี

ข้อดีของแบบจำลองฐานข้อมูลเชิงวัตถุ คือ แบบจำลองฐานข้อมูลเชิงวัตถุสามารถเก็บข้อมูลที่ซับซ้อนและมีความยืดหยุ่นสูง โดยสามารถเก็บข้อมูลที่มีคุณลักษณะและพฤติกรรมที่หลากหลายในวัตถุเดียวกัน และสามารถสร้างวัตถุใหม่ที่สืบทอดคุณสมบัติจากวัตถุเดิมได้

2) ข้อจำกัด

ข้อจำกัดของแบบจำลองฐานข้อมูลเชิงวัตถุ คือ ความซับซ้อนในการออกแบบและการใช้งาน การออกแบบและการใช้งานฐานข้อมูลเชิงวัตถุต้องใช้ความรู้และทักษะทางเชิงวัตถุ การเข้าใจและออกแบบสัมพันธ์ระหว่างวัตถุอาจเป็นภาระงานที่ซับซ้อนและต้องใช้ความชำนาญเพื่อทำได้ถูกต้อง อีกทั้ง ความยากในการแสดงความสัมพันธ์ระหว่างข้อมูล การสืบทอดและการสร้างความสัมพันธ์ระหว่างวัตถุอาจทำให้ข้อมูลซับซ้อนและยากต่อการจัดการ อาจเกิดปัญหาในการค้นหาและอัพเดตข้อมูลที่เกี่ยวข้องกัน



ภาพที่ 5.6 แบบจำลองฐานข้อมูลเชิงวัตถุ

2.5 แบบจำลองฐานข้อมูลแบบมัลติไดเมนชั่น (Multidimensional Database Model)

แบบจำลองฐานข้อมูลแบบมัลติไดเมนชั่นเป็นแบบจำลองที่ใช้ในการจัดเก็บและจัดการข้อมูลที่มีลักษณะเป็นข้อมูลหลายมิติ แทนที่จะใช้โครงสร้างแบบตารางที่ใช้ในแบบจำลองเชิงสัมพันธ์ ข้อมูลจะถูกจัดเก็บในรูปแบบของแกน (Axis) และมีลักษณะเป็นตารางมิติ (Dimensional Table) ที่ประกอบด้วยแกนและเซลล์ข้อมูล แบบจำลองนี้เหมาะสำหรับระบบฐานข้อมูลที่มีการวิเคราะห์ข้อมูลที่มีมิติมาก แบบจำลองรูปแบบนี้ใช้งานกับคลังข้อมูล (Data Warehousing) เช่น ระบบฐานข้อมูลสนับสนุนการตัดสินใจ ระบบวิเคราะห์ข้อมูลทางธุรกิจ และระบบวิเคราะห์ข้อมูลทางการเงิน แสดงดังภาพ 5.7 สำหรับข้อดีและข้อจำกัดของแบบจำลองรูปแบบนี้ มีดังนี้

1) ข้อดี

ข้อดีของแบบจำลองฐานข้อมูลแบบมัลติไดเมนชั่น คือ การวิเคราะห์และสืบค้นข้อมูลที่มีความซับซ้อน เป็นระบบฐานข้อมูลที่ใช้ในการวิเคราะห์ข้อมูลที่มีความซับซ้อน และสามารถสืบค้นข้อมูลได้อย่างมีประสิทธิภาพ อีกทั้ง ยังมีความยืดหยุ่นในการออกแบบและการใช้งานด้วย

2) ข้อจำกัด

ข้อจำกัดของแบบจำลองฐานข้อมูลแบบมัลติไดเมนชั่น คือ การออกแบบและการใช้งานฐานข้อมูลรูปแบบนี้ต้องใช้ความรู้และทักษะทางการวิเคราะห์แบบมิติ การเข้าใจและออกแบบตารางข้อมูลและมิติอาจเป็นภาระงานที่ซับซ้อนและต้องใช้ความชำนาญเพื่อทำได้ถูกต้อง และไม่เหมาะสมสำหรับข้อมูลที่มีการ เปลี่ยนแปลงบ่อย เนื่องจากการเพิ่มหรือลบข้อมูลในมิติอาจต้องการการปรับปรุงโครงสร้างข้อมูลที่ซับซ้อนและเป็นเวลานาน

ภาพที่ 5.7 แบบจำลองฐานข้อมูลแบบมัลติไดเมนชั่น

ระบบจัดการฐานข้อมูล (Database Management System : DBMS)

ระบบจัดการฐานข้อมูล (Database Management System) หรือเรียกคำย่อว่า DBMS คือ ซอฟต์แวร์ที่ใช้ในการจัดการและดำเนินการกับฐานข้อมูล เพื่อเก็บรักษาข้อมูล การเข้าถึงข้อมูล และการจัดการโครงสร้างข้อมูลให้เป็นไปตามความต้องการของผู้ใช้งาน DBMS มีฟังก์ชันและคุณสมบัติหลายอย่างที่ช่วยให้ผู้ใช้สามารถจัดการฐานข้อมูลได้อย่างมีประสิทธิภาพ โดยเน้นความสามารถในการเก็บรักษาข้อมูลที่ปลอดภัย การจัดการโครงสร้างฐานข้อมูล เช่น การสร้าง อ่าน แก้ไข และลบข้อมูล การจัดการความสัมพันธ์ระหว่างตาราง และการจัดการความมั่นคงและความปลอดภัยของข้อมูล DBMS มีประโยชน์หลายด้าน เช่น ช่วยให้การจัดเก็บข้อมูลมีความปลอดภัย ช่วยให้ผู้ใช้สามารถเข้าถึงข้อมูลได้ง่ายและรวดเร็ว รองรับการแบ่งปันข้อมูลในทีมและระหว่างหน่วยงาน และช่วยในการสำรองข้อมูลและกู้คืนข้อมูลเมื่อมีปัญหาเกิดขึ้น

1. ส่วนประกอบด้านสภาพแวดล้อมของระบบจัดการฐานข้อมูล

ระบบจัดการฐานข้อมูลประกอบด้วยส่วนประกอบด้านสภาพแวดล้อมที่สำคัญ เช่น ฮาร์ดแวร์ ซอฟต์แวร์ ข้อมูล โพรซีเซส และผู้ใช้งานส่วนประกอบเหล่านี้ทำงานร่วมกันเพื่อให้ระบบจัดการฐานข้อมูลทำงานได้อย่างมีประสิทธิภาพ และผู้ใช้งานสามารถเข้าถึงและจัดการข้อมูลในระบบได้อย่างถูกต้องและปลอดภัย สามารถอธิบายแต่ละส่วนได้ ดังนี้

1.1 ฮาร์ดแวร์ (Hardware) เป็นส่วนที่เกี่ยวข้องกับอุปกรณ์ที่ใช้ในการเก็บรักษาและประมวลผลข้อมูลในระบบฐานข้อมูล เช่น เซิร์ฟเวอร์ (servers) เครื่องคอมพิวเตอร์ และอุปกรณ์เกี่ยวข้องอื่น ๆ เช่น ฮาร์ดดิสก์ (hard disks) เครือข่าย (network) และอุปกรณ์เกี่ยวกับการสำรองข้อมูล (backup devices) ฮาร์ดแวร์จะ

รับผิดชอบในการเก็บรักษาข้อมูลในรูปแบบที่ปลอดภัยและมีประสิทธิภาพเพื่อให้ระบบจัดการฐานข้อมูลทำงานได้อย่างเต็มที่

1.2 ซอฟต์แวร์ (Software) เป็นส่วนที่รับผิดชอบในการจัดการและประมวลผลข้อมูลในระบบฐานข้อมูล ซอฟต์แวร์ประกอบด้วยระบบจัดการฐานข้อมูล (Database Management System: DBMS) ที่รับผิดชอบในการเก็บรักษาและจัดการข้อมูล รวมถึงภาษาควบคุมฐานข้อมูล (Database Control Language: DCL) และภาษาสอบถามข้อมูล (Data Query Language: DQL) เช่น SQL ที่ใช้ในการสร้างและจัดการข้อมูลในฐานข้อมูล

1.3 ข้อมูล (Data) เป็นส่วนที่สำคัญที่สุดของระบบจัดการฐานข้อมูล ซึ่งเป็นข้อมูลที่ถูกจัดเก็บในระบบฐานข้อมูล ข้อมูลเก็บรวบรวมข้อมูลที่เกี่ยวข้องกันและถูกออกแบบให้เข้าถึงและประมวลผลได้อย่างมีประสิทธิภาพ รูปแบบของข้อมูลอาจเป็นตาราง (tables) ลำดับชั้น (hierarchical) วัตถุ (objects) หรืออื่น ๆ ขึ้นอยู่กับแบบจำลองฐานข้อมูลที่ใช้

1.4 โพรซีเยอร์ (Procedure) เป็นส่วนที่รับผิดชอบในการประมวลผลข้อมูลในระบบฐานข้อมูล โดยใช้โครงสร้างและลำดับขั้นตอนที่กำหนดเพื่อให้ระบบดำเนินการตามคำสั่งที่กำหนด โพรซีเยอร์จะเป็นส่วนสำคัญในการควบคุมกระบวนการที่เกิดขึ้นในระบบฐานข้อมูล เช่น การเพิ่มหรือลบข้อมูล การสร้างรายงาน หรือการปรับปรุงโครงสร้างข้อมูล เป็นต้น

1.5 ใช้งาน (Users) เป็นส่วนที่สำคัญในการใช้งานระบบจัดการฐานข้อมูล ผู้ใช้งานสามารถเป็นผู้ใช้ทั่วไปหรือผู้ใช้ที่มีความเชี่ยวชาญในการจัดการฐานข้อมูล พวกเขาสามารถใช้งานระบบฐานข้อมูลเพื่อค้นหาข้อมูล แก้ไขข้อมูล สร้างรายงาน หรือดำเนินการอื่น ๆ ที่เกี่ยวข้องกับข้อมูลในระบบ

2. บทบาทหน้าที่ของผู้ที่เกี่ยวข้องในระบบฐานข้อมูล

บทบาทหน้าที่ของผู้ที่เกี่ยวข้องในระบบฐานข้อมูลเป็นสิ่งสำคัญ เนื่องจากการวางแผนและออกแบบระบบฐานข้อมูลที่ดีและเหมาะสม รวมทั้งการดูแลและปรับปรุงระบบอย่างต่อเนื่องเป็นปัจจัยสำคัญที่มีผลต่อประสิทธิภาพและประสิทธิผลของระบบฐานข้อมูลในองค์กร และให้บริการที่ตอบสนองต่อความต้องการและความพึงพอใจของผู้ใช้งาน สามารถแบ่งกลุ่มหน้าที่ของผู้เกี่ยวข้องออกเป็น 4 ตำแหน่งหลัก ๆ คือ (Connolly & Begg, 2014)

2.1 ผู้บริหารข้อมูลและผู้บริหารฐานข้อมูล (Data and Database Administrators)

บทบาทหน้าที่ของผู้บริหารข้อมูลและผู้บริหารฐานข้อมูล (Database Administrators : DBA) คือการวางแผน ดูแล และควบคุมระบบฐานข้อมูลทั้งหมด มีความรับผิดชอบในการกำหนดและดูแลนโยบายความปลอดภัยของข้อมูล การประสานงานกับผู้ใช้งานและนักพัฒนา การจัดการสิทธิ์การเข้าถึงข้อมูล การสำรองข้อมูล และการปรับปรุงประสิทธิภาพของระบบฐานข้อมูลอย่างต่อเนื่อง

2.2 นักออกแบบฐานข้อมูล (Database Designers)

นักออกแบบฐานข้อมูลมีหน้าที่ออกแบบโครงสร้างและโมเดลข้อมูลที่เหมาะสมสำหรับฐานข้อมูลในระบบ ตำแหน่งนี้ต้องเข้าใจความต้องการของผู้ใช้งานและวิเคราะห์ข้อมูลที่ต้องการจัดเก็บ รวมทั้งเลือกใช้รูปแบบข้อมูลที่เหมาะสม เช่น ระบบสนับสนุนการตัดสินใจ (Decision Support System) หรือระบบการทำงานที่ต้องการความรวดเร็ว (Real-time System)

2.3 นักวิเคราะห์ระบบ (System Analysis)

นักวิเคราะห์ระบบเป็นผู้ที่รับผิดชอบในการวิเคราะห์ความต้องการของผู้ใช้งานและการกำหนดความเหมาะสมของระบบฐานข้อมูล ตำแหน่งนี้ทำหน้าที่ในการเก็บข้อมูล วิเคราะห์และออกแบบโครงสร้างระบบฐานข้อมูล รวมทั้งจัดการการทดสอบ การปรับปรุงแก้ไข และการจัดการเกี่ยวกับการนำระบบฐานข้อมูลไปใช้งานจริง

2.4 ผู้ใช้งาน (End-Users)

ผู้ใช้งาน คือบุคคลหรือองค์กรที่ใช้งานระบบฐานข้อมูลในการเข้าถึงและใช้งานข้อมูล มีหลากหลายรูปแบบขึ้นอยู่กับบทบาทและความต้องการของผู้ใช้งาน อาจเป็นผู้ใช้ทั่วไปที่ใช้เครื่องมือในการสืบค้นและอ่านข้อมูล หรือผู้ใช้งานที่มีความเชี่ยวชาญในการสืบค้นและวิเคราะห์ข้อมูลเพื่อการตัดสินใจ

ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

1. โครงสร้างฐานข้อมูลเชิงสัมพันธ์ (Relational Database Structure)

โครงสร้างฐานข้อมูลเชิงสัมพันธ์ (Relational Database Structure) เป็นโครงสร้างที่นำมาใช้ในระบบจัดการฐานข้อมูลแบบเชิงสัมพันธ์ ซึ่งจะแสดงข้อมูลในรูปแบบของตารางที่มีความสัมพันธ์กัน โดยมีส่วนประกอบที่สำคัญดังนี้ (Elmasri, R., & Navathe, S. B., 2015)

1) รีเลชั่น (Relation) เป็นคอนเซ็ปต์พื้นฐานของโครงสร้างฐานข้อมูลเชิงสัมพันธ์ หรือตารางข้อมูลที่ประกอบด้วยแถวและคอลัมน์ แต่ละตารางแสดงข้อมูลเกี่ยวกับแอตทริบิวต์ที่ต้องการจัดเก็บ

2) แอตทริบิวต์ (Attribute) แสดงถึงคุณลักษณะหรือข้อมูลที่เกี่ยวข้องกับแต่ละตาราง แต่ละแอตทริบิวต์จะมีชื่อและประเภทข้อมูลที่กำหนด เช่น ชื่อ ที่อยู่ เบอร์โทรศัพท์ เป็นต้น

3) โดเมน (Domain) เป็นเซตของค่าที่สามารถเก็บได้ในแอตทริบิวต์แต่ละตัว โดเมนกำหนดชนิดข้อมูลและขอบเขตของข้อมูลที่อนุญาต เช่น โดเมนของแอตทริบิวต์เงินเดือนสามารถเป็นจำนวนเต็มบวก

4) ทูเพิล (Tuple) แสดงถึงแถวของข้อมูลในตาราง แต่ละทูเพิลจะเป็นชุดของค่าที่เกี่ยวข้องกับแอตทริบิวต์แต่ละตัว ในทูเพิลสามารถเก็บข้อมูลของแอตทริบิวต์ที่กำหนดได้ ยกตัวอย่างเช่น ในตารางข้อมูลลูกค้า แต่ละทูเพิลจะมีข้อมูลเกี่ยวกับชื่อลูกค้า ที่อยู่ เบอร์โทรศัพท์ เป็นต้น

5) ดีกรี (Degree) แสดงถึงจำนวนแอตทริบิวต์ที่อยู่ในตาราง หรือจำนวนคอลัมน์ในตาราง ซึ่งเป็นตัวบ่งบอกถึงความซับซ้อนของโครงสร้างฐานข้อมูล ตารางที่มีดีกรีสูงก็จะมีจำนวนแอตทริบิวต์มากขึ้น

6) คาร์ดินัลลิตี้ (Cardinality) แสดงถึงจำนวนทูเพิลที่อยู่ในตาราง หรือจำนวนแถวในตาราง ค่าคาร์ดินัลลิตี้สามารถใช้บอกถึงขนาดของตารางฐานข้อมูล

2. คุณสมบัติของรีเลชัน (Properties of Relations)

2.1 คุณสมบัติของรีเลชัน ประกอบด้วย

1) รีเลชันต้องมีชื่อกำกับ และแต่ละรีเลชันต้องมีชื่อแตกต่างกัน ซ้ำกันไม่ได้

2) แต่ละแอตทริบิวต์ของรีเลชันจะต้องมีค่าเพียงค่าเดียว (Atomic) กล่าวคือ ค่าที่บรรจุนั้นจะไม่สามารถแบ่งย่อยได้อีก

3) แต่ละแอตทริบิวต์ในรีเลชันนั้นๆ ต้องมีชื่อแตกต่างกัน จะมีชื่อแอตทริบิวต์ซ้ำกันไม่ได้

4) ค่าต่าง ๆ ทั้งหมดที่บรรจุลงในแอตทริบิวต์หนึ่ง ๆ ต้องเป็นไปตามข้อกำหนดของโดเมนเดียวกัน

5) แต่ละทูเพิลต้องมีข้อมูลแตกต่างกัน ซ้ำกันไม่ได้

6) ลำดับของแต่ละแอตทริบิวต์ ไม่มีความสำคัญใด ๆ

7) ในเชิงทฤษฎี การจัดเรียงลำดับของทูเพิล ไม่มีความสำคัญใด ๆ (แต่มุมมองเชิงปฏิบัติแล้วการจัดเรียงลำดับของทูเพิลส่งผลต่อประสิทธิภาพในการเข้าถึงข้อมูล)

2.2 คีย์ (Keys)

ข้อมูลที่จัดเก็บอยู่ในฐานข้อมูลเชิงสัมพันธ์ จะถูกนำเสนอออกมาในรูปแบบของตาราง 2 มิติ ที่ประกอบไปด้วยแถวและคอลัมน์ และด้วยคุณสมบัติของรีเลชันตามที่ได้กล่าวไว้แล้วในข้อที่ 5 คือ “แต่ละทูเพิลต้องมี ข้อมูลแตกต่างกัน ซ้ำกันไม่ได้” ดังนั้นจึงจำเป็นต้องมีการกำหนดแอตทริบิวต์ หรือกลุ่มของแอตทริบิวต์ที่เรียกว่า “คีย์” เพื่อนำไปใช้ระบุความเป็นเอกลักษณ์ (Uniquely) ของแต่ละทูเพิลได้ (คีย์จะซ้ำกันไม่ได้) กล่าวคือ การ กำหนดคีย์ให้กับรีเลชัน ทำให้เราสามารถนำคีย์เหล่านี้ไประบุทูเพิลในรีเลชันตามที่เราต้องการได้นั่นเอง โดยคีย์ที่ใช้งานในฐานข้อมูล ประกอบด้วยคีย์ชนิดต่างๆ ดังต่อไปนี้ (โอภาส เอี่ยมสิริวงศ์, 2558)

1) ซูเปอร์คีย์ (Super key)

ซูเปอร์คีย์ คือ แอตทริบิวต์หรือกลุ่มของแอตทริบิวต์ที่ใช้ระบุถึงความเป็นเอกลักษณ์ในแต่ละทูเพิลของ รีเลชันนั้นๆ ได้ ดังนั้น ในรีเลชันหนึ่งๆ จึงสามารถมีซูเปอร์คีย์ได้หลายตัว ยกตัวอย่างเช่น รีเลชัน EMPLOYEE ที่ประกอบไปด้วยแอตทริบิวต์ต่างๆ ดังต่อไปนี้

EMPLOYEE (empNo, empName, empPosition)

และจากนิยามของซูเปอร์คีย์ เราจึงสามารถกำหนดซูเปอร์คีย์ให้กับรีเลชัน EMPLOYEE เพื่อใช้ระบุความแตกต่างหรือความเป็นเอกลักษณ์ของแต่ละทูเพิลได้ ดังนี้

(empNo) (empNo, empName) (empName)

2) คีย์คู่แข่ง (Candidate Key: CK)

ในรีเลชันหนึ่ง ๆ อาจมีคีย์คู่แข่งได้หลายตัว โดยคีย์คู่แข่งก็คือซูเปอร์คีย์ที่มีขนาดเล็กที่สุดที่สามารถระบุความแตกต่างในแต่ละทูเพิลบนรีเลชันได้ กล่าวคือเป็นซูเปอร์คีย์ที่ไม่มีซับเซตของตัวเองที่นำมาใช้เป็นซูเปอร์คีย์นั่นเอง ดังนั้น กลุ่มของแอตทริบิวต์ (empNo, empName) คีย์เหล่านี้จึงไม่ถือว่าเป็นคีย์คู่แข่ง เนื่องจากมี ซับเซตเป็นคีย์คู่แข่ง ดังนั้น รีเลชัน EMPLOYEE จึงมีแอตทริบิวต์ที่เหลือคือ empNo และ empName ที่ สามารถนำมาใช้เป็นคีย์คู่แข่งได้

3) คีย์หลัก (Primary Key: PK)

คีย์หลักคือคีย์คู่แข่งที่ถูกคัดเลือกให้เป็นคีย์หลัก เพื่อนำไปอ้างอิงความเป็นเอกลักษณ์ของแต่ละทูเพิลใน รีเลชันนั้น ๆ ได้คีย์หลักต้องไม่บรรจุค่าว่าง (Null) โดยในที่นี้ได้คัดเลือกแอตทริบิวต์ empNo เป็นคีย์หลัก ดังนั้นคีย์คู่แข่งที่ไม่ได้ผ่านการคัดเลือก ก็จะเรียกว่าคีย์รอง (Alternate Key/Secondary Key) ซึ่งคีย์รองเมื่อนำไป

ค้นหาข้อมูลจากความสัมพันธ์ อาจได้ทูเพิลมากกว่าหนึ่งเรคอร์ดนั่นเป็นเพราะว่าคีย์รองนั้นไม่มีความเป็นเอกลักษณ์ ตัวอย่างเช่น ให้แอตทริบิวต์ empName เป็นคีย์รอง ดังนั้น ในตารางพนักงานอาจมีพนักงานที่มีชื่อเหมือนกันก็ได้

4) คีย์นอก (Foreign Key: FK)

คีย์นอกประกอบด้วยแอตทริบิวต์หรือกลุ่มของแอตทริบิวต์ในรีเลชันหนึ่งที่มีคุณสมบัติเป็นคีย์หลักและไปปรากฏในอีกรีเลชันหนึ่งเพื่อใช้สำหรับเชื่อมโยงระหว่างกัน คีย์นอกจัดเป็นคีย์ที่มีความสำคัญมากในฐานข้อมูลเชิงสัมพันธ์ เนื่องจากเป็นคีย์ที่ถูกนำไปใช้เพื่อการเชื่อมโยงความสัมพันธ์ระหว่างตาราง รีเลชัน BRANCH จะมีคีย์หลักคือแอตทริบิวต์ branchNo ในขณะที่รีเลชัน EMPLOYEE จะมีแอตทริบิวต์ branchNo เช่นกัน แต่ถูกกำหนดเป็นคีย์นอกเพื่อใช้เชื่อมโยงไปยังรีเลชัน BRANCH นั่นเอง

นอกจากนี้ ยังมีคีย์ที่ประกอบด้วยกลุ่มของแอตทริบิวต์ ที่เรียกว่า คีย์รวม (Composite Key) ซึ่งคีย์ดังกล่าวยังสามารถเรียกได้อีกหลายชื่อด้วยกัน เช่น Concatenated Key หรือ Compound Key

แบบจำลอง E-R (Entity-Relational Modeling)

แบบจำลอง E-R (Elmasri, R., & Navathe, S. B., 2016) ถูกพัฒนาขึ้นในปี ค.ศ. 1976 โดย Peter Chen เพื่อใช้ในการออกแบบและวางแผนระบบฐานข้อมูลเชิงสัมพันธ์ แบบจำลองนี้ใช้สัญลักษณ์และแผนภาพในการแสดงความสัมพันธ์ระหว่างเอ็นทิตี้และแอตทริบิวต์ในฐานข้อมูล และได้รับความนิยมในวงกว้างเนื่องจากความคล้ายคลึงกับแนวคิดและวิธีการคิดของมนุษย์ในการวางแผนและสื่อสาร (โอภาส เอี่ยมสิริวงศ์, 2558)

แบบจำลอง E-R จัดเป็นแบบจำลองเชิงแนวคิด (Conceptual Data Model) ที่ใช้แสดงลักษณะโดยรวมของข้อมูลในระบบ เป็นแบบจำลองที่ใช้ในการวางแผนและออกแบบระบบฐานข้อมูลเชิงสัมพันธ์ โดยใช้แผนภาพหรือไดอะแกรมที่ช่วยในการแสดงความสัมพันธ์ระหว่างแอตทริบิวต์และเอ็นทิตี้ โดยทั่วไปแล้วผู้ออกแบบฐานข้อมูลจะใช้แบบจำลอง E-R เพื่อเข้าใจและกำหนดคุณสมบัติของระบบฐานข้อมูลที่ต้องการสร้างขึ้น

ในแบบจำลอง E-R นักออกแบบฐานข้อมูลจะใช้สัญลักษณ์และแผนภาพเพื่ออธิบายส่วนประกอบต่างๆ ในระบบฐานข้อมูล หลักการทำงานของแบบจำลอง E-R คือ การระบุแอตทริบิวต์ที่อธิบายคุณสมบัติหรือลักษณะของเอ็นทิตี้แต่ละตัว และการกำหนดความสัมพันธ์ระหว่างเอนทิตีที่เกี่ยวข้องกัน โดยสามารถใช้ความสัมพันธ์หลายรูปแบบได้ เช่น ความสัมพันธ์หนึ่งต่อหลาย (one-to-many) ความสัมพันธ์หลายต่อหลาย (many-to-many) เป็นต้น

แบบจำลอง E-R ช่วยให้ผู้ออกแบบฐานข้อมูลสามารถสื่อสารและเข้าใจความต้องการของผู้ใช้งานและรายละเอียดของฐานข้อมูลได้ง่ายขึ้น แผนภาพ E-R ที่สร้างขึ้นจะแสดงข้อมูลที่มีอยู่ในธุรกิจหรือระบบที่กำลังออกแบบ รวมถึงเอ็นทิตี้และแอตทริบิวต์ที่เกี่ยวข้อง เช่น แผนภาพ E-R จะแสดงว่ามี Entity ใดอยู่ในระบบ และมี Attribute อะไรบ้าง รวมทั้งแสดงความสัมพันธ์ระหว่าง Entity ในรูปแบบใด ผู้ใช้งานที่ไม่มีความรู้ด้านเทคนิคก็สามารถเข้าใจแผนภาพ E-R ได้อย่างง่าย นอกจากนี้ บทบาทของผู้ใช้งานก็เป็นสำคัญในการให้ข้อมูลแก่นักออกแบบฐานข้อมูลเพื่อพัฒนาแผนภาพ E-R ให้ครบถ้วนด้วย

ในการที่จะสร้างแผนภาพ E-R ได้นั้นจำเป็นต้องเรียนรู้ถึงสัญลักษณ์ที่ใช้แทนความหมายต่างๆ ให้เข้าใจ ซึ่งประกอบด้วยรายละเอียดที่สำคัญดังต่อไปนี้ (โอภาส เอี่ยมสิริวงศ์, 2558)

1. เอ็นทิตี้ (Entity)

เอนทิตี คือ วัตถุที่เราสนใจซึ่งอาจเป็นได้ทั้ง บุคคล สถานที่ วัตถุ เหตุการณ์ หรือแนวคิดที่ก่อให้เกิดกลุ่มของข้อมูลที่ต้องการ และต่อไปนี้เป็นตัวอย่างของเอนทิตีต่าง

บุคคล เช่น Employee (พนักงาน) Student (นักศึกษา) Patient (ผู้ป่วย)

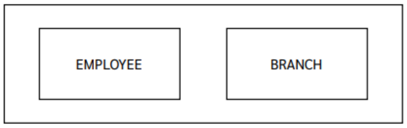
สถานที่ เช่น Store (ร้านค้า) Warehouse (คลังสินค้า) Station (สถานี)

วัตถุ เช่น Machine (เครื่องจักร) Building (ตึกอาคาร) Automobile (เครื่องยนต์)

เหตุการณ์ เช่น Sale (การขาย) Registration (การลงทะเบียน) Reservation (การจองห้องพัก)

แนวคิด เช่น Account (บัญชี) Course (หลักสูตร) Work Center (ศูนย์การทำงาน)

สัญลักษณ์เอ็นทิตี้จะเป็นรูปสี่เหลี่ยมผืนผ้าดังภาพที่ 5.8 สำหรับการตั้งชื่อให้กับเอ็นทิตี้นั้นชื่อที่ใช้จะต้องเป็นคำนาม และโดยปกติมักจะเขียนด้วยตัวพิมพ์ใหญ่

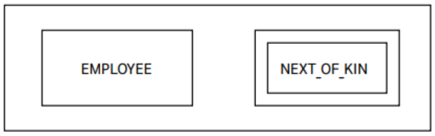


ภาพที่ 5.8 สัญลักษณ์เอ็นทิตี้

นอกจากนี้ เอ็นทิตี้ยังสามารถแบ่งออกเป็น 2 ประเภท คือ

1. Strong Entity คือ เป็นเอ็นทิตี้ที่มีคุณสมบัติเป็นตัวต้นแบบและเป็นอิสระไม่ขึ้นกับเอ็นทิตี้ใดๆ โดยมีการระบุคุณสมบัติที่เป็นระเบียบเรียง (Key Attribute) ที่สามารถระบุและแยกแยะเอนทิตีแต่ละตัวจากกันได้อย่างชัดเจน ตัวอย่างเอ็นทิตี้ Strong เป็นเอ็นทิตี้ที่มีความสำคัญในระบบฐานข้อมูล เช่น ลูกค้า (Customer) หรือสินค้า (Product) ซึ่งมีข้อมูลที่เป็นเอ็นทิตี้ Strong เองและสามารถระบุและแยกแยะจากเอ็นทิตี้อื่นๆ ในระบบได้ โดยใช้รหัสลูกค้าหรือรหัสสินค้าเป็นตัวระบุ (Key) เพื่อสร้างความสัมพันธ์กับเอ็นทิตี้อื่น ๆ ในระบบฐานข้อมูล

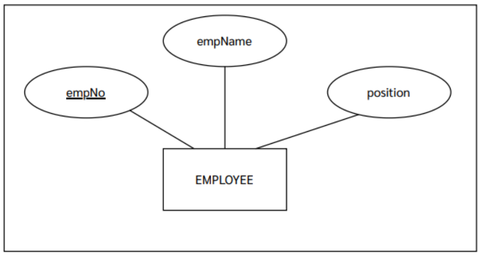
2. Weak Entity คือ เป็นเอนทิตีอ่อนแอที่ต้องพึ่งพาเอ็นทิตี้อื่น ๆ เป็นตัวระบุ เพื่อสร้างความสัมพันธ์และติดต่อกับเอ็นทิตี้ Strong ตัวอย่างเอ็นทิตี้อ่อนแอเป็นตัวที่มีความสัมพันธ์กับเอ็นทิตี้ Strong และเกี่ยวข้องกับเวลา เช่น รายการสั่งซื้อ (Order) ที่มีรหัสรายการสั่งซื้อเป็นรหัสลูกค้าและเป็นตัวระบุที่เชื่อมโยงกับลูกค้าเพื่อแสดงความสัมพันธ์ ดังภาพที่ 3.9



ภาพที่ 5.9 Strong Entityและ Weak Entity

2. แอตทริบิวต์ (Attributes)

เป็นส่วนประกอบสำคัญในแบบจำลอง E-R ที่ใช้ในการกำหนดคุณลักษณะหรือลักษณะเฉพาะของเอ็นทิตี้ (Entity) ในระบบฐานข้อมูล แอตทริบิวต์เป็นข้อมูลที่ใช้ในการบ่งชี้คุณลักษณะเฉพาะของเอ็นทิตี้ สัญลักษณ์แอตทริบิวต์จะเป็นรูปวงรี แอตทริบิวต์ที่ถูกใช้เป็นคีย์หลักจะถูกขีดเส้นใต้กำกับไว้ เช่น เอนทิตี Employee ประกอบไปด้วยแอตทริบิวต์ emp_no, emp_name และ position โดยกำหนดให้แอตทริบิวต์ emp_no เป็นคีย์หลัก สามารถเขียนได้ดังภาพที่ 5.10

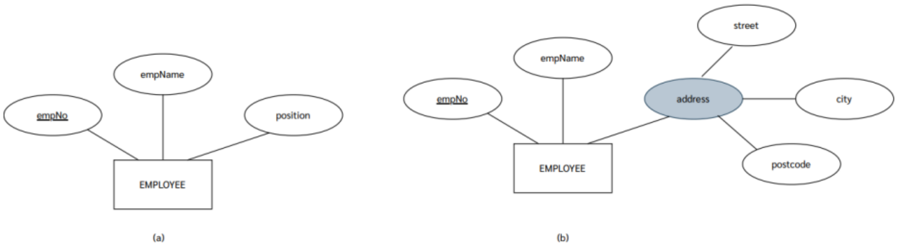


ภาพที่ 5.10 เอ็นทิตี้ Employee ซึ่งประกอบด้วยแอตทริบิวต์ต่างๆ

นอกจากนี้ แอตทริบิวต์ยังสามารถแบ่งออกเป็นหลายชนิดด้วยกัน คือ

2.1 Attribute Domain เป็นการกำหนดขอบเขตคำข้อมูล และชนิดของข้อมูลให้กับแอตทริบิวต์นั่นหมายถึง โดเมน เป็นตัวกำหนดความเป็นไปได้ของข้อมูลในแอตทริบิวต์นั้น ๆ ตัวอย่างเช่น Attribute Domain ของแอตทริบิวต์ sex จะมีค่าที่เป็นไปได้คือ M หรือ F ได้เท่านั้น เป็นต้น

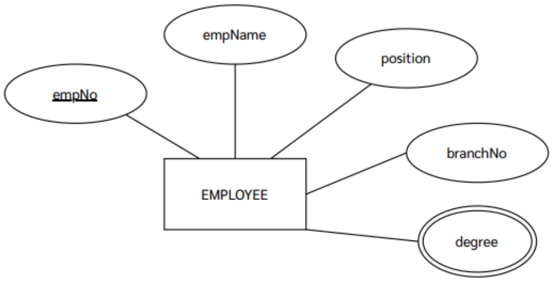
2.2 Simple Attributeและ Composite Attribute คือแอตทริบิวต์ที่มีองค์ประกอบเดียว กล่าวคือ เป็นแอตทริบิวต์ที่ไม่สามารถแตกเป็นข้อมูลย่อยได้อีก ตัวอย่างเช่น แอตทริบิวต์ age, sex และ salary เป็นต้น ในขณะที่ Composite Attribute คือแอตทริบิวต์ที่มีองค์ประกอบอยู่หลายตัว มีข้อมูลมากกว่าหนึ่งค่าภายในแอตทริบิวต์นั้น ตัวอย่างเช่น แอตทริบิวต์ชื่อ address ที่ภายในประกอบด้วยแอตทริบิวต์ street, city และ postcode ดังภาพที่ 5.11



ภาพที่ 5.11 เปรียบเทียบระหว่าง Simple Attribute และ Composite Attribute

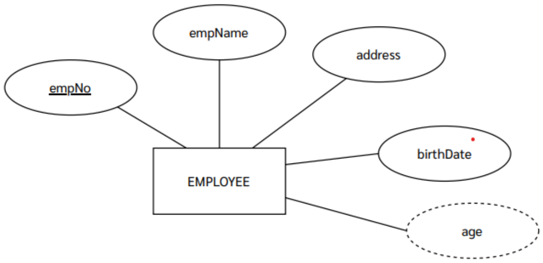
2.3 Single-Valued Attribute และ Multivalued Attribute คือ แอตทริบิวต์ที่บรรจุค่าเพียงค่าเดียว ตัวอย่างเช่น พนักงานต้องทำงานอยู่ในสาขา ดังนั้น แอตทริบิวต์ branchNo ในเอ็นทิตี้ EMPLOYEE จึงสามารถนำไปอ้างอิงข้อมูลสาขาได้เพียงหนึ่งสาขาเท่านั้น ในขณะที่ Multivalued Attibute คือแอตทริบิวต์ที่สามารถมีหลายค่าได้ ตัวอย่างเช่น พนักงานคนหนึ่งสามารถมีวุฒิการศึกษาได้หลายระดับ หรือภายในครอบครัว

หนึ่งสามารถมีเบอร์โทรศัพท์ได้หลายเบอร์ เป็นต้น โดยสัญลักษณ์ Multivalued Attribute เส้นวงรีจะเป็นเส้นคู่ เขียนได้ดังภาพที่ 5.12



ภาพที่ 5.12 Single-Valued Attributeและ Multivalued Attribute

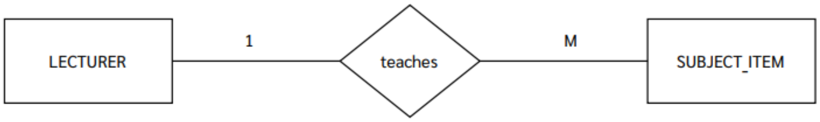
2.4 Derived Attribute คือแอตทริบิวต์ที่เกิดจากการประยุกต์ด้วยแอตทริบิวต์อื่น เช่น แอตทริบิวต์ age ที่ได้มาจากการนำแอตทริบิวต์วันเกิด (birthdate) มาคำนวณใช้งานด้วยการนำปีปัจจุบันหักด้วยปีเกิดจะได้แอตทริบิวต์ age โดยสัญลักษณ์ของแอตทริบิวต์นี้จะเป็นวงรีแบบเส้นปะ และยังสามารถเรียกอีกชื่อหนึ่งว่า Computed Attribute ดังภาพที่ 5.13



ภาพที่ 5.13 แอตทริบิวต์ age เป็น Derived Attribute

3. ความสัมพันธ์ (Relationship)

ความสัมพันธ์ในที่นี้หมายถึงความความสัมพันธ์ระหว่างเอ็นทิตี้ โดยแต่ละความสัมพันธ์ควรมีชื่อระบุไว้เพื่อใช้อธิบาย ซึ่งปกติจะใช้สัญลักษณ์สี่เหลี่ยมขนมเปียกปูน (◇) ที่ภายในระบุคำกริยาไว้เพื่ออธิบายความสัมพันธ์ ตัวอย่างเช่น อาจารย์ สอนหนังสือได้หลายวิชา เป็นต้น เขียนได้ดังภาพที่ 5.14

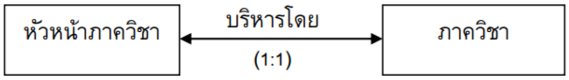


ภาพที่ 5.14 แสดงการอธิบายความสัมพันธ์ระหว่างอาจารย์และรายวิชาที่สอน

ความสัมพันธ์ของเอ็นทิตี้จะเกิดขึ้นได้ 3 รูปแบบ

1) ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-One Relationships)

เอ็นทิตี้ A มีความสัมพันธ์แบบหนึ่งต่อหนึ่งกับเอ็นทิตี้ B หมายถึง ข้อมูลหนึ่งแถวใด ๆ ของ A จะมีความสัมพันธ์กับข้อมูล B เพียงแถวเดียวเท่านั้น สามารถเขียนเป็นความสัมพันธ์ในวงเล็บได้ว่า (1:1) แสดงให้เห็นว่าเป็นความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อหนึ่ง ดังภาพที่ 5.15

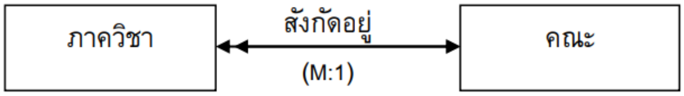


ภาพที่ 5.15 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (1:1)

ความสัมพันธ์ระหว่างเอ็นทิตี้ของภาพที่ 5.15 หมายถึง หัวหน้าภาควิชา 1 คน บริหารได้ 1 ภาควิชา ในขณะที่ทางกลับกัน 1 ภาควิชา บริหารโดย หัวหน้าภาควิชา 1 คน จะเขียนเส้นความสัมพันธ์ตรงกลางได้ว่า (1:1) ซึ่งหัวลูกศรมีความหมายว่า ถ้าเป็น 1 หัว หมายถึง เป็น One

2) ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-to-Many Relationships)

เอ็นทิตี้ A มีความสัมพันธ์แบบหนึ่งต่อหนึ่งกลุ่มกับเอ็นทิตี้ B หมายถึง ข้อมูล 1 แถวใน A สามารถมีความสัมพันธ์กับข้อมูลใน B ได้มากกว่า 1 แถว แต่เมื่อมองย้อนกลับข้อมูล 1 แถวใน B จะมีความสัมพันธ์กับข้อมูลใน A ได้เพียงหนึ่งแถว

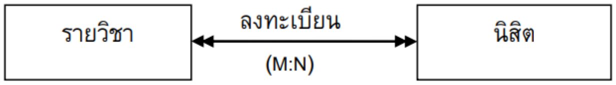


ภาพที่ 5.16 ความสัมพันธ์ระหว่างเอ็นทิตี้แบบ One-to-Many

จากภาพที่ 5.16 หมายถึง ภาควิชาสังกัดอยู่ได้เพียง 1 คณะในขณะที่ทางกลับกัน 1 คณะมีได้หลายภาควิชา อาจเกิดข้อสงสัยว่าทำไมขึ้นด้วย Many ก่อนและตามด้วย One ซึ่งในความเป็นจริงแล้วไม่จำเป็นว่าจะอยู่ด้านใดไม่ว่าจะเป็น 1:M หรือ M:1 เมื่ออธิบายความสัมพันธ์แล้วมีค่าเท่ากัน เพียงแต่ใครจะเป็น One หรือ เป็น Many เท่านั้น ไม่มีความแตกต่างกันนั่นเอง

3. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to-Many Relationships)

เอ็นทิตี้ A มีความสัมพันธ์แบบกลุ่มต่อกลุ่มกับเอ็นทิตี้ B หมายถึงข้อมูลแต่ละแถวใน A มีความสัมพันธ์กับข้อมูลใน B ได้มากกว่า 1 แถว และเมื่อมองย้อนกลับข้อมูลแต่ละแถวใน B ก็สามารถมีความสัมพันธ์กับข้อมูลใน A ได้มากกว่า 1 แถว เช่นเดียวกัน จะเขียนความสัมพันธ์ ระหว่างเอ็นทิตี้ แทนด้วย (M:N) ดังภาพที่ 5.17 เช่น

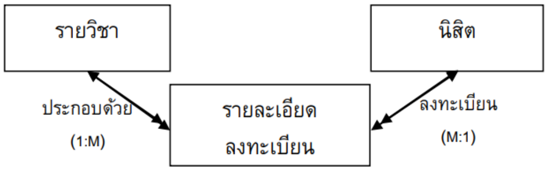


ภาพที่ 5.17 ความสัมพันธ์ระหว่างเอ็นทิตี้แบบ Many-to-Many

จากภาพที่ 5.17 ความสัมพันธ์ระหว่างเอนทิตีแบบ Many-to-Many หมายถึง รายวิชาหลายรายวิชาที่นิสิตลงทะเบียนได้มากกว่า 1 รายวิชา ในขณะที่นิสิตหลายคนก็สามารถลงทะเบียนได้หลาย

รายวิชาซึ่งความสัมพันธ์ระหว่าง 2 เอ็นทิตี้ที่เป็นแบบกลุ่มต่อกลุ่ม นั้นจะไม่เกิดขึ้น เนื่องจากจะเกิดความยุ่งยากที่จะไปกระทบกับเอ็นทิตี้ตัวอื่นด้วย เช่น อาจมีปัญหาในด้านการปรับปรุงแก้ไขข้อมูล โดยทั่วไปมักจะไม่ยอมให้มีเอ็นทิตี้ที่มีความสัมพันธ์กันแบบกลุ่มต่อกลุ่มในฐานข้อมูล (ยกเว้นฐานข้อมูลแบบ Network Model)

วิธีการปรับแก้ไม่ให้เกิด Many to Many คือ ให้สร้างเอนทิตีขึ้นมาใหม่ที่เป็นตัวเชื่อมระหว่างสองเอ็นทิตี้เดิม จะเรียกว่า “Associative Entity” และปรับความสัมพันธ์ของเอ็นทิตี้เดิมกับใหม่ให้อยู่ในรูป 1:M ได้ดังนี้



ภาพที่ 5.18 ความสัมพันธ์ระหว่างเอนทิตีแบบ One-to-Many ที่แปลงจาก Many-to-Many

ความสัมพันธ์ระหว่างเอ็นทิตี้ของภาพที่ 5.18 หมายถึง ข้อมูลรายวิชา ประกอบด้วยรายละเอียดของรายวิชา เช่น รหัสรายวิชา ชื่อวิชา จำนวนหน่วยกิต ข้อมูลนิสิตที่ลงทะเบียน ประกอบด้วยรายละเอียดของนิสิต เช่น รหัสนิสิต ชื่อนิสิต นามสกุลนิสิต สาขาวิชา นั่นคือ ในรายละเอียดลงทะเบียนจะเป็นตัวเชื่อมความสัมพันธ์ของสองเอ็นทิตี้เดิม และดึงคีย์หลัก (Primary Key: PK) ของทั้งสองเอ็นทิตี้มาไว้ที่เอ็นทิตี้ใหม่ ซึ่งก็จะสามารถเรียกใช้ข้อมูลได้ทั้งสองเอ็นทิตี้เดิมนั่นเอง

กระบวนการในการพัฒนาแผนภาพ E-R ในองค์กรธุรกิจสมัยใหม่

กระบวนการในการพัฒนาแผนภาพ E-R (Entity-Relational Modeling) ในองค์กรธุรกิจสมัยใหม่เป็นกระบวนการที่เน้นการทวนซ้ำ (Iteration) และปรับปรุงเพื่อให้แผนภาพที่สร้างขึ้นมีความชัดเจนและครอบคลุมความต้องการของระบบฐานข้อมูลที่เกี่ยวข้องกับธุรกิจนั้น ๆ กระบวนการนี้สามารถเข้าใจได้โดยแบ่งเป็นขั้นตอนดังนี้

1. การเขียนรายละเอียดการปฏิบัติงานขององค์กรเป็นข้อความบรรยาย เริ่มต้นด้วยการเข้าใจและบ่งบอกรายละเอียดของการทำงานในองค์กร โดยระบุความต้องการและปัญหาที่ต้องการแก้ไขด้วยข้อความที่ชัดเจนและเข้าใจได้ง่าย

2. การระบุรายละเอียดกฎเกณฑ์ของทางธุรกิจ หลังจากเข้าใจรายละเอียดของการทำงานในขั้นตอนที่ 1 จะมีการระบุกฎเกณฑ์หรือกลุ่มกฎเกณฑ์ที่เกี่ยวข้องกับธุรกิจ ซึ่งเป็นเกณฑ์ที่ช่วยกำหนดขอบเขตและเงื่อนไขในการออกแบบฐานข้อมูล

3. การกำหนดเอนทิตีและความสัมพันธ์จากกฎเกณฑ์ทางธุรกิจ จากกฎเกณฑ์ที่กำหนดมา จะมีการกำหนดแอตทริบิวต์และความสัมพันธ์ระหว่างเอ็นทิตี้ต่าง ๆ ที่เกี่ยวข้องกับธุรกิจ

4. เขียนแผนภาพ E-R ฉบับเริ่มต้น จากแอตทริบิวต์และความสัมพันธ์ที่กำหนดมา จะมีการสร้างแผนภาพ E-R ในรูปแบบเริ่มต้น ซึ่งแสดงความสัมพันธ์ระหว่างเอ็นทิตี้ต่าง ๆ และแอตทริบิวต์ที่เกี่ยวข้อง

5. กำหนดแอตทริบิวต์ที่เพียงพอต่อการอธิบายรายละเอียดของเอ็นทิตี้รวมถึงกำหนดคีย์หลัก จากแผนภาพ E-R ฉบับเริ่มต้น จะตรวจสอบและปรับปรุงแอตทริบิวต์ที่เพียงพอต่อการอธิบายรายละเอียดของเอ็นทิตี้รวมถึงกำหนดแอตทริบิวต์ที่เป็นคีย์หลักที่ใช้ในการระบุเอ็นทิตี้แต่ละตัว

6. ทบทวน ปรับปรุงและตรวจสอบแผนภาพ E-R ใหม่ ในกระบวนการนี้ จะมีการทบทวนและปรับปรุงแผนภาพ E-R เพื่อให้ครอบคลุมและเข้าใจได้ถูกต้องตามความต้องการของระบบ ซึ่งการทบทวนและปรับปรุงนี้จะเกี่ยวข้องกับการเพิ่มเติม เอ็นทิตี้ แอตทริบิวต์ และการเชื่อมโยงความสัมพันธ์ และจะทวนซ้ำไปเรื่อย ๆ จนกว่าผู้ใช้งานและผู้ออกแบบจะตกลงกันว่าแผนภาพ E-R ที่สร้างขึ้นมีความชัดเจนและครอบคลุมกิจกรรมและฟังก์ชั่นของธุรกิจอย่างเหมาะสม

กระบวนการดังกล่าวเป็นการวางแผนและปรับปรุงแผนภาพ E-R ให้เข้ากับความต้องการของระบบฐานข้อมูลและธุรกิจสมัยใหม่ และเป็นกระบวนการที่ทวนซ้ำเพื่อให้ได้แผนภาพ E-R ที่ถูกต้องและสอดคล้องกับความต้องการขององค์กรธุรกิจในปัจจุบัน

การเขียนคำสั่ง SQL จัดการฐานข้อมูล

ภาษา SQL (Structured Query Language) เป็นภาษามาตรฐานที่ใช้ในการจัดการและเข้าถึงข้อมูลในระบบฐานข้อมูลเชิงสัมพันธ์ ภาษา SQL ถือเป็นภาษาคิวรี (Query Language) ที่มีความสามารถในการสอบถาม (Query) และแก้ไขข้อมูลในฐานข้อมูล

ประวัติภาษา SQL เริ่มต้นขึ้นในปี ค.ศ. 1970 โดย Dr. Edgar F. Codd ผู้มีผลงานด้านระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งได้แนะนำโครงสร้างข้อมูลที่เรียกว่าระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database System) ที่มีความสามารถในการจัดเก็บข้อมูลและสัมพันธ์กันในรูปแบบตาราง โครงสร้างข้อมูลที่มีความยืดหยุ่นและมีประสิทธิภาพนี้ได้รับความนิยมอย่างรวดเร็วในอุตสาหกรรมฐานข้อมูล

SQL ได้รับการพัฒนาและพัฒนาต่อยอดในระยะเวลาหลายสิบกว่าปี กลายเป็นภาษามาตรฐานที่ใช้ในการจัดการฐานข้อมูลทั่วไป โดยมีรายละเอียดและความสามารถที่รองรับการสร้างตาราง การเพิ่ม-ลบ-แก้ไขข้อมูล การสอบถามข้อมูล การสร้างคำสั่งเงื่อนไข การจัดกลุ่มข้อมูล และการรวมข้อมูลจากหลายตาราง

ภาษา SQL เป็นภาษาที่เข้าใจง่ายและมีการใช้งานที่แพร่หลายในวงกว้าง นอกจากนี้ยังมีฐานข้อมูลที่รองรับภาษา SQL อย่างมากมาย เช่น MySQL, Oracle, Microsoft SQL Server, PostgreSQL เป็นต้น ทำให้ SQL เป็นเครื่องมือที่สำคัญในการจัดการข้อมูลในองค์กรและธุรกิจในปัจจุบัน

1. ประเภทของคำสั่ง SQL

ประเภทของคำสั่ง SQL แบ่งออกเป็น 3 ประเภท ดังนี้

1. ภาษานิยามข้อมูล (Data Definition Language : DDL)

เป็นชุดคำสั่งที่ใช้ในการสร้างและจัดการโครงสร้างข้อมูลในฐานข้อมูล เช่น การสร้างตาราง (CREATE TABLE), การเปลี่ยนแปลงโครงสร้างตาราง (ALTER TABLE) การลบตาราง (DROP TABLE) เป็นต้น ภาษา DDL ช่วยให้ผู้ใช้สามารถกำหนดคุณสมบัติและโครงสร้างข้อมูลในฐานข้อมูลได้ตามต้องการ ภาษา DDL มีบทบาทสำคัญในการสร้างและบริหารจัดการโครงสร้างข้อมูลให้เป็นไปตามความต้องการขององค์กร

2. ภาษาจัดการข้อมูล (Data Manipulation Language : DML)

เป็นชุดคำสั่งที่ใช้ในการจัดการข้อมูลในฐานข้อมูล เช่น การเพิ่มข้อมูลเข้าสู่ตาราง (INSERT INTO) การอัปเดตข้อมูลในตาราง (UPDATE) การลบข้อมูลจากตาราง (DELETE FROM) เป็นต้น ภาษา DML ช่วยให้ผู้ใช้สามารถเพิ่ม แก้ไข และลบข้อมูลในตารางได้อย่างมีประสิทธิภาพ ภาษา DML เป็นเครื่องมือสำคัญในการจัดการและปรับเปลี่ยนข้อมูลในฐานข้อมูล

3. ภาษาควบคุมข้อมูล (Data Control Language : DCL)

เป็นชุดคำสั่งที่ใช้ในการควบคุมสิทธิ์และการเข้าถึงข้อมูลในฐานข้อมูล เช่น การกำหนดสิทธิ์การเข้าถึง (GRANT) การเพิกถอนสิทธิ์การเข้าถึง (REVOKE) เป็นต้น ภาษา DCL ช่วยให้ผู้ดูแลระบบสามารถกำหนดสิทธิ์การเข้าถึงข้อมูลและควบคุมการเข้าถึงข้อมูลในระบบฐานข้อมูลได้

2. การเขียนคำสั่ง SQL

2.1 คำสั่งภาษานิยามข้อมูล (DDL)

เป็นประเภทของคำสั่ง SQL ที่ใช้ในการสร้างและกำหนดโครงสร้างข้อมูลในฐานข้อมูล โดยภาษา DDL ช่วยให้ผู้ใช้สามารถสร้างตาราง (Table) และกำหนดคุณสมบัติของตารางได้อย่างชัดเจน ยกตัวอย่างคำสั่ง ดังนี้

1) CREATE คือ คำสั่งใช้สร้างวัตถุในฐานข้อมูล เช่น ตาราง (TABLE) วิว (VIEW) และ ดัชนี (INDEX) เป็นต้น สามารถเขียนคำสั่งได้ ดังนี้

คำสั่ง CREATE SCHEMA ใช้สร้าง schema (โครงสร้างฐานข้อมูล) ในฐานข้อมูล Schema เป็นกลุ่มของตาราง วิว ฟังก์ชัน และอ็อบเจกต์อื่น ๆ ที่เกี่ยวข้องกันภายในฐานข้อมูล ซึ่งชื่อของ schema จะใช้เพื่อแยกแยะและจัดการกับวัตถุในฐานข้อมูล โครงสร้างคำสั่ง ดังนี้

```
CREATE SCHEMA schema_name;
```

ยกตัวอย่าง คำสั่ง CREATE SCHEMA ใช้สร้าง schema ชื่อ "my_schema" ในฐานข้อมูล ดังนี้

```
CREATE SCHEMA my_schema;
```

คำสั่ง CREATE DOMAIN ใช้สร้าง domain (โดเมน) ซึ่งเป็นการกำหนดชนิดข้อมูลที่กำหนดเองในฐานข้อมูล Domain ช่วยให้สามารถกำหนดข้อกำหนดเพิ่มเติมและความถูกต้องของข้อมูลในตารางได้ โครงสร้างคำสั่ง ดังนี้

```
CREATE DOMAIN domain_name AS data_type;
```

ยกตัวอย่างคำสั่ง CREATE DOMAIN ใช้สร้าง domain ชื่อ "age_domain" ที่เก็บข้อมูลประเภท INTEGER ในฐานข้อมูล ดังนี้

```
CREATE DOMAIN age_domain AS INTEGER;
```

คำสั่ง CREATE TABLE ใช้สร้างตาราง (table) ในฐานข้อมูล โดยระบุชื่อตารางและประกอบด้วยคอลัมน์ที่ต้องการสร้าง และประเภทข้อมูลของแต่ละคอลัมน์ โครงสร้างคำสั่ง ดังนี้

```
CREATE TABLE table_name (
   column1 data_type,
   column2 data_type,
   ...
);
```

ยกตัวอย่างคำสั่ง CREATE TABLE ใช้สร้างตาราง "employees" ที่ประกอบด้วยคอลัมน์ employee_id, first_name, last_name, และ department_id ในฐานข้อมูล ดังนี้ โครงสร้างคำสั่งดังนี้

```
CREATE TABLE employees (
   employee_id INT,
   first_name VARCHAR(50),
   last_name VARCHAR(50),
   department_id INT
);
```

คำสั่ง CREATE VIEW ใช้สร้างวิว (view) ซึ่งเป็นตารางที่สร้างจากการสืบค้นข้อมูลจากตารางหรือวิวอื่น ๆ ในฐานข้อมูล วิวเป็นแบบสมมาตรกับข้อมูลที่อยู่ในตารางและมีการปรับปรุงข้อมูลแบบอัตโนมัติ

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

ยกตัวอย่างคำสั่ง CREATE VIEW ใช้สร้างวิว "sales_summary" ที่เป็นผลลัพธ์จากการสร้างรายงานยอดขายโดยรวมจากตาราง "sales" ในฐานข้อมูล โดยรวมข้อมูลโดยใช้ฟังก์ชัน SUM และกลุ่มข้อมูลตามชื่อสินค้า (product_name) ดังนี้

```
CREATE VIEW sales_summary AS
SELECT product_name, SUM(quantity) AS total_quantity
FROM sales
GROUP BY product_name;
```

2) ALTER คือ คำสั่งใช้เปลี่ยนแปลงโครงสร้างของวัตถุในฐานข้อมูล เช่น เพิ่มหรือลบคอลัมน์ในตาราง เปลี่ยนประเภทข้อมูลของคอลัมน์ เป็นต้น สามารถเขียนคำสั่งได้ดังนี้

ALTER DOMAIN ใช้เปลี่ยนการตั้งค่าเริ่มต้น (default value) ของ domain ในฐานข้อมูล โดยระบุชื่อของ domain ที่ต้องการเปลี่ยนในส่วน domain_name และระบุค่าเริ่มต้นที่ต้องการใหม่ในส่วน default_value โครงสร้างคำสั่ง ดังนี้

```
ALTER DOMAIN domain_name SET DEFAULT default_value;
```

ยกตัวอย่างคำสั่ง ALTER DOMAIN ใช้เปลี่ยนคุณสมบัติของ domain ที่มีชื่อว่า "age_domain" ในฐานข้อมูล ในตัวอย่างนี้ คำสั่ง SET NOT NULL ถูกใช้เพื่อกำหนดให้คอลัมน์ที่ใช้ domain ต้องไม่เป็นค่าว่าง (NULL) ดังนี้

```
ALTER DOMAIN age_domain SET NOT NULL;
```

ALTER TABLE ใช้เพิ่มคอลัมน์ใหม่ในตารางในฐานข้อมูล โดยระบุชื่อของตารางที่ต้องการเพิ่มคอลัมน์ในส่วน table_name และระบุชื่อของคอลัมน์และประเภทข้อมูลใหม่ในส่วน column_name และ data_type โครงสร้างคำสั่ง ดังนี้

```
ALTER TABLE table_name ADD COLUMN column_name data_type;
```

ยกตัวอย่างคำสั่ง ALTER TABLE ใช้เพิ่มคอลัมน์ในตาราง "employees" ในฐานข้อมูล ในตัวอย่างนี้ คอลัมน์ที่ชื่อว่า "salary" ถูกเพิ่มโดยระบุประเภทข้อมูลเป็น DECIMAL(10,2) ซึ่งแสดงถึงตัวเลขที่มี 10 ตำแหน่งทั้งหมดและตำแหน่งทศนิยม 2 ตำแหน่ง ดังนี้

```
ALTER TABLE employees ADD COLUMN salary DECIMAL(10,2);
```

3) DROP คือ คำสั่งใช้ลบวัตถุในฐานข้อมูล เช่น ลบตาราง (TABLE) ลบวิว (VIEW) ลบดัชนี (INDEX) เป็นต้น สามารถเขียนคำสั่งได้ดังนี้

DROP SCHEMA คือคำสั่งใช้ในการลบ schema (โครงสร้างระดับสูง) จากฐานข้อมูล โครงสร้างคำสั่ง ดังนี้

```
DROP SCHEMA schema_name;
ตัวอย่าง
DROP SCHEMA my_schema;
```

DROP DOMAIN ใช้ในการลบ domain (ประเภทข้อมูลที่กำหนดเอง) จากฐานข้อมูล โครงสร้างคำสั่ง ดังนี้

```
DROP DOMAIN domain_name;
ตัวอย่าง
DROP DOMAIN age_domain;
```

DROP TABLE ใช้ในการลบตาราง (โครงสร้างข้อมูล) จากฐานข้อมูล โดยต้องระบุชื่อตารางที่ต้องการลบ โครงสร้างคำสั่ง ดังนี้

```
DROP TABLE table_name;
ตัวอย่าง
DROP TABLE employees;
```

DROP VIEW ใช้ในการลบวิว (ตารางเสมือน) จากฐานข้อมูล โดยต้องระบุชื่อวิวที่ต้องการลบโครงสร้างคำสั่ง ดังนี้

```
DROP VIEW view_name;
ตัวอย่าง
DROP TABLE sales_summary ;
```

2.2 คำสั่งภาษาจัดการข้อมูล (DML)

เป็นคำสั่งที่ใช้ในการเพิ่ม แก้ไข ลบ และค้นหาข้อมูลในฐานข้อมูล เพื่อจัดการข้อมูลในตาราง โดยมีคำสั่งหลัก ได้แก่ INSERT, UPDATE, DELETE, และ SELECT สามารถเขียนคำสั่งได้ ดังนี้

1) INSERT คำสั่งใช้ในการเพิ่มข้อมูลใหม่ลงในตารางข้อมูล โดยระบุค่าข้อมูลที่ต้องการเพิ่มในแต่ละคอลัมน์ของตาราง โครงสร้างคำสั่ง ดังนี้

```
INSERT INTO table_name (column1, column2, ...)
VALUES (value1, value2, ...);
```

ยกตัวอย่างคำสั่ง INSERT เพิ่มข้อมูลใหม่ลงในตาราง employees โดยกำหนดค่าของ employee_id เป็น 1001, first_name เป็น 'John', last_name เป็น 'Doe', และ department_id เป็น 101 ดังนี้

```
INSERT INTO employees (employee_id, first_name, last_name, department_id)
VALUES (1001, 'John', 'Doe', 101);
```

2) UPDATE ใช้ในการปรับปรุงข้อมูลในตาราง โดยระบุคอลัมน์ที่ต้องการปรับปรุงและค่าที่ต้องการแทนที่ในคอลัมน์นั้น โครงสร้างคำสั่ง ดังนี้

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

ยกตัวอย่าง อัปเดตค่าในคอลัมน์ department_id เป็น 101 สำหรับแถวที่มีค่า employee_id เป็น 1001 ในตาราง employees ดังนี้

```
UPDATE employees
SET department_id = 101
WHERE employee_id = 1001;
```

3) DELETE ใช้ในการลบข้อมูลออกจากตาราง โดยระบุเงื่อนไขว่าข้อมูลใดที่ต้องการลบออก โครงสร้างคำสั่ง ดังนี้

```
DELETE FROM table_name
WHERE condition;
```

ยกตัวอย่าง ลบแถวที่มีค่า employee_id เป็น 1001 ออกจากตาราง employees

```
DELETE FROM employees
WHERE employee_id = 1001;
```

4) SELECT ใช้ในการเรียกดูข้อมูลจากตาราง โดยระบุคอลัมน์ที่ต้องการและเงื่อนไขที่ต้องการใช้ในการกรองข้อมูล โครงสร้างคำสั่ง ดังนี้

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

ยกตัวอย่าง เรียกดูข้อมูลคอลัมน์ first_name และ last_name จากตาราง employees สำหรับแถวที่มีค่า department_id เป็น 101 ดังนี้

```
SELECT first_name, last_name
FROM employees
WHERE department_id = 101;
```

2.3 คำสั่งภาษาควบคุมข้อมูล (DCL)

เป็นชุดคำสั่งที่ใช้ในการควบคุมสิทธิ์และการเข้าถึงข้อมูลในฐานข้อมูล เช่น การกำหนดสิทธิ์การเข้าถึง (GRANT) การเพิกถอนสิทธิ์การเข้าถึง (REVOKE) เป็นต้น สามารถเขียนคำสั่งได้ ดังนี้

1) GRANT ใช้ในการกำหนดสิทธิ์การเข้าถึงและการดำเนินการต่าง ๆ ในฐานข้อมูลให้กับผู้ใช้หรือบทบาทผู้ใช้ โครงสร้างคำสั่ง ดังนี้

```
GRANT privileges
ON object
TO user;
```

ตัวอย่าง การกำหนดสิทธิ์ให้ผู้ใช้ user1 สามารถเรียกดูข้อมูล (SELECT) เพิ่มข้อมูล (INSERT) และอัปเดตข้อมูล (UPDATE) ในตาราง employees ดังนี้

```
GRANT SELECT, INSERT, UPDATE
ON employees
TO user1;
```

2) REVOKE ใช้ในการเพิกถอนสิทธิ์การเข้าถึงและการดำเนินการต่าง ๆ ในฐานข้อมูลจากผู้ใช้หรือบทบาทผู้ใช้ โครงสร้างคำสั่ง ดังนี้

```
REVOKE privileges
ON object
FROM user;
```

ตัวอย่าง การเพิกถอนสิทธิ์ให้ผู้ใช้ user1 ไม่สามารถเรียกดูข้อมูล (SELECT) เพิ่มข้อมูล (INSERT) และอัปเดตข้อมูล (UPDATE) ในตาราง employees ได้อีกต่อไป

```
REVOKE SELECT, INSERT, UPDATE
ON employees
FROM user1;
```

สรุปท้ายบทที่ 5

การออกแบบฐานข้อมูลเป็นกระบวนการสำคัญที่มีความสำคัญในการจัดการข้อมูลในองค์กรธุรกิจ โดยมุ่งเน้นในการสร้างโครงสร้างและองค์ประกอบของระบบฐานข้อมูลที่เหมาะสมและมีประสิทธิภาพในการบริหารจัดการข้อมูล ปัญหาเกี่ยวกับแฟ้มข้อมูลและการใช้ระบบแฟ้มข้อมูลในอดีตได้รับการแก้ไขด้วยการใช้ฐานข้อมูล ทำให้มีความเป็นระเบียบและเพิ่มประสิทธิภาพในการเก็บรวบรวมและเข้าถึงข้อมูล

ในการออกแบบฐานข้อมูล จะใช้แบบจำลอง E-R เพื่อเข้าใจและแสดงความสัมพันธ์ระหว่างเอ็นทิตี้ที่เกี่ยวข้องในระบบฐานข้อมูล โดยนำโครงสร้างและแบบจำลอง E-R มาสร้างเป็นโครงสร้างฐานข้อมูลเชิงสัมพันธ์ โดยใช้คำสั่ง SQL เพื่อสร้างตาราง (CREATE TABLE) และกำหนดความสัมพันธ์ระหว่างตารางต่าง ๆ โดยมีองค์ประกอบที่เกี่ยวข้อง ได้แก่ คีย์หลัก (Primary Key) และคีย์นอก (Foreign Key)

ระบบจัดการฐานข้อมูล (DBMS) เป็นซอฟต์แวร์ที่ใช้ในการจัดการฐานข้อมูล ซึ่งรวมถึงการสร้างและจัดการฐานข้อมูล รวมถึงการเข้าถึงและปรับปรุงข้อมูล โดยอาศัยภาษา SQL เป็นภาษาในการจัดการฐานข้อมูล

โครงสร้างฐานข้อมูลเชิงสัมพันธ์เป็นโครงสร้างที่ได้รับความนิยมในการจัดเก็บข้อมูล โดยใช้ตารางที่เกี่ยวข้องกัน และใช้คีย์หลักและคีย์ต่อใช้ในการเชื่อมโยงข้อมูลระหว่างตาราง และแบบจำลอง E-R เป็นเครื่องมือที่ใช้ในการออกแบบและอธิบายโครงสร้างของข้อมูลในฐานข้อมูล โดยทำให้เราสามารถระบุและเข้าใจความสัมพันธ์ระหว่างเอนทิตีในระบบฐานข้อมูลได้อย่างชัดเจน

การเขียนคำสั่ง SQL เป็นการใช้ภาษาคอมพิวเตอร์ที่เข้าใจง่ายและใช้ในการจัดการฐานข้อมูล ได้แก่ การเพิ่ม (INSERT) ข้อมูลใหม่ การแก้ไข (UPDATE) ข้อมูลที่มีอยู่ การลบ (DELETE) ข้อมูลที่ไม่ต้องการ และการเลือก (SELECT) ข้อมูลที่ต้องการแสดงผล

ดังนั้น การออกแบบฐานข้อมูลเป็นกระบวนการที่ซับซ้อนและมีความสำคัญในการจัดการข้อมูลในองค์กรธุรกิจ ซึ่งคำนึงถึงโครงสร้างข้อมูลที่เหมาะสม แบบจำลองที่เข้าใจง่าย รวมถึงการใช้เครื่องมือ DBMS และคำสั่ง

คำถามท้ายบทที่ 5

1. ปัญหาเกี่ยวกับแฟ้มข้อมูลมีอะไรบ้าง จงอธิบาย
2. จงอธิบายข้อแตกต่างระหว่างการจัดการข้อมูลด้วยแฟ้มข้อมูลกับวิธีฐานข้อมูล
3. จงอธิบายสภาพแวดล้อมของ DBMS
4. สรุปข้อดีและข้อจำกัดของฐานข้อมูล
5. แบบจำลองฐานข้อมูลมีกี่รูปแบบ อะไรบ้าง พร้อมสรุปแต่ละรูปแบบ มาพอสังเขป
6. จงอธิบายส่วนต่าง ๆ ของโครงสร้างฐานข้อมูลเชิงสัมพันธ์
7. จงอธิบายความสัมพันธ์แบบหนึ่งต่อหนึ่ง พร้อมยกตัวอย่าง
8. จงอธิบายความสัมพันธ์แบบหนึ่งต่อกลุ่ม พร้อมยกตัวอย่าง
9. จงอธิบายความสัมพันธ์แบบกลุ่มต่อกลุ่ม พร้อมยกตัวอย่าง
10. ประเภทของคำสั่ง SQL มีกี่ประเภท จงอธิบาย

บทที่ 6
การออกแบบส่วนติดต่อผู้ใช้

การออกแบบส่วนติดต่อผู้ใช้ (User Interface Design) เป็นกระบวนการที่สำคัญในการพัฒนาระบบที่ต้องให้ผู้ใช้และระบบงานสามารถสื่อสารและโต้ตอบกันได้อย่างมีประสิทธิภาพ การออกแบบส่วนติดต่อผู้ใช้เกี่ยวข้องกับการสร้างสื่อประสานกับผู้ใช้เพื่อให้ผู้ใช้สามารถเข้าถึงและใช้งานระบบได้อย่างง่าย มีประสิทธิภาพและเหมาะสมต่อการทำงานของผู้ใช้แต่ละคน ผู้ใช้แต่ละคนมีพื้นฐานการใช้เครื่องคอมพิวเตอร์ที่แตกต่างกัน หากผู้ใช้มีพื้นฐานที่มากและเข้าใจหลักการทำงานของระบบคอมพิวเตอร์มาก่อน จะทำให้การดำเนินงานในระบบเป็นไปอย่างราบรื่นและมีประสิทธิภาพมากที่สุด การออกแบบส่วนติดต่อผู้ใช้ควรใช้ภาษาที่เข้าใจง่ายและสื่อความหมายได้ถูกต้องตามฟังก์ชันการทำงานของระบบ เพื่อให้ผู้ใช้สามารถใช้งานได้ง่ายและมีประสิทธิภาพ การออกแบบส่วนติดต่อผู้ใช้ให้เข้าใจและสื่อสารกับระบบคอมพิวเตอร์เป็นเรื่องสำคัญ เนื่องจากการสื่อสารที่เข้าใจกันอย่างถูกต้องจะช่วยให้ผู้ใช้มีประสบการณ์การใช้งานที่ดีและเป็นที่พอใจ นอกจากนี้ยังมีการใช้กราฟิกเพื่อสร้างความสวยงามและน่าสนใจเพื่อสร้างความพึงพอใจและความสนุกสนานในการใช้งาน

ก่อนหน้านี้การทำงานของส่วนประสานผู้ใช้เป็นเรื่องที่ลำบาก ผู้ใช้งานจะต้องมีความรู้เกี่ยวกับคำสั่งในการทำให้โปรแกรมทำงานตามที่เราต้องการ รูปแบบการแสดงผลขึ้นอยู่กับโปรแกรม เช่น การใช้งานโปรแกรมภาษาซี (C programming) ลักษณะการติดต่อกับโปรแกรมและผู้ใช้งานนั้น ผู้จะป้อนคำสั่งลงไปในระบบทีละคำสั่ง และการประมวลผลก็จะประมวลผลคำสั่งเหล่านั้นทีละบรรทัด ในสภาพแวดล้อมนี้เกิดการใช้งานแบบ Command-Line Interface (CLI) ซึ่งทำให้การใช้งานโปรแกรมเป็นไปยากมากขึ้น การพัฒนาส่วนการติดต่อผู้ใช้งานตั้งแต่อดีตจนถึงปัจจุบันมีการพัฒนาอย่างต่อเนื่องเพื่อให้สอดคล้องกับพฤติกรรมการใช้งานของผู้ใช้มากขึ้น เราสามารถเห็นถึงตัวอย่างที่ชัดเจนได้ในส่วนการติดต่อผู้ใช้ของระบบปฏิบัติการ Mac ที่ถูกออกแบบให้มีการใช้งานที่สะดวกสบายสำหรับผู้ใช้ รูปแบบกราฟิกของส่วนการติดต่อก็มีจุดเด่นและลักษณะเฉพาะที่แตกต่างจากระบบปฏิบัติการ Windows ตัวอย่างเช่นการเปิด-ปิดหน้าต่างที่อยู่ทางขวาใน Windows และอยู่ทางซ้ายใน Mac การออกแบบและดีไซน์ของแถบการใช้งาน (Taskbar) ก็มีเอกลักษณะเฉพาะของ Mac ซึ่งเป็นผู้นำทางด้านกราฟิกและส่วนการติดต่อผู้ใช้ที่ดีและเหมาะสมต่อการใช้งาน

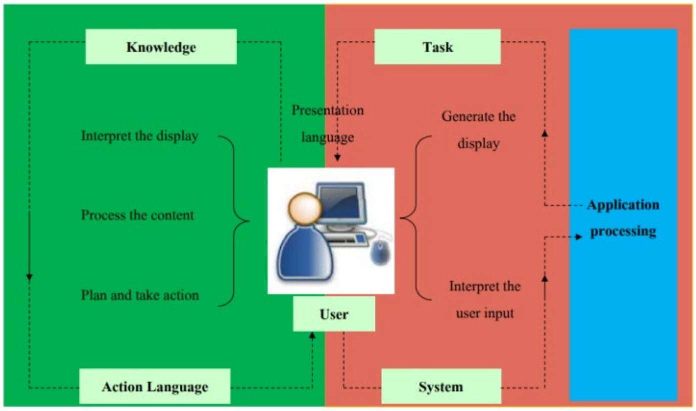
ดังนั้น การออกแบบส่วนการติดต่อผู้ใช้งานเป็นสิ่งสำคัญอย่างยิ่งในการพัฒนาโปรแกรมประยุกต์ ไม่เพียงแต่การออกแบบให้ผู้ใช้งานสามารถทำงานได้อย่างรวดเร็ว แต่ยังเน้นให้ผู้ใช้งานสามารถใช้งานโปรแกรมได้อย่างมีประสิทธิภาพมากขึ้น การออกแบบไอคอน หน้าต่าง และการทำงานต่าง ๆ ของโปรแกรมมีความสำคัญที่

นักพัฒนาควรให้ความสำคัญ แนวทางการออกแบบที่ดีอย่างเช่นในหลักการออกแบบของจาคอป (Jakop's Principle of Design) เป็นตัวอย่างของแนวทางที่น่าสนใจที่นักพัฒนาโปรแกรมสามารถอ้างอิงและนำไปใช้ได้

องค์ประกอบของส่วนติดต่อผู้ใช้

ผู้ใช้โปรแกรมคอมพิวเตอร์ส่วนใหญ่ต้องการโปรแกรมที่มีการออกแบบที่ไม่ซับซ้อนและเข้าใจง่าย เนื่องจากความรู้และประสบการณ์ในการใช้เครื่องคอมพิวเตอร์ไม่มากนัก อย่างไรก็ตามเทคโนโลยีปัจจุบันได้พัฒนาและเติบโตอย่างรวดเร็ว นักพัฒนาโปรแกรมและระบบคอมพิวเตอร์สามารถใช้เครื่องมือที่มีความสะดวกมากขึ้นในการสร้างส่วนติดต่อที่ใช้งานง่ายและสามารถสนับสนุนการแสดงผลแบบกราฟิกได้มากขึ้น ผู้ใช้งานสามารถใช้คอมพิวเตอร์ได้โดยไม่จำเป็นต้องมีความรู้ลึกเกี่ยวกับคอมพิวเตอร์มาก่อนหน้านี้ ความต้องการแบบนี้ทำให้มีการเรียนรู้และการสนับสนุนในการออกแบบส่วนติดต่อกับผู้ใช้มากขึ้นเรื่อย ๆ ในอุตสาหกรรมการพัฒนาโปรแกรมคอมพิวเตอร์ (โอภาส เอี่ยมสิริวงศ์, 2556)

สื่อประสานกับผู้ใช้จึงมีส่วนสำคัญต่อผู้ใช้งานระบบในการดำเนินการและตัดสินใจ โดยทั่วไปผู้บริหารที่ไม่ใช่ผู้เชี่ยวชาญในเรื่องคอมพิวเตอร์ ดังนี้ การออกแบบส่วนการจัดการสื่อประสานกับผู้ใช้ ควรเริ่มจากการศึกษาถึงองค์ประกอบของสื่อประสานกับผู้ใช้ ดังภาพที่ 6.1



ภาพที่ 6.1 องค์ประกอบของสื่อประสานกับผู้ใช้

จากภาพในที่นี้จะแบ่งองค์ประกอบของสื่อประสานกับผู้ใช้ออกเป็น 2 กลุ่ม ได้แก่ กลุ่มผู้ใช้และกลุ่มระบบคอมพิวเตอร์ แต่ละกลุ่มจะมีองค์ประกอบย่อยดังรายละเอียดต่อไปนี้

1. กลุ่มผู้ใช้ระบบ

1) องค์ความรู้ (Knowledge) หมายถึง ความรู้ของผู้ใช้ระบบที่จะต้องมีหรือจะต้องรู้ไม่ว่าจะเป็นความรู้เรื่องใดๆ ที่จะนำมาใช้เพื่อติดต่อกับระบบคอมพิวเตอร์ องค์ความรู้นี้อาจเกิดจากความเชี่ยวชาญของผู้ใช้ระบบเองหรืออาจเกิดจากการเรียนรู้จากแหล่งอื่น ๆ การมีองค์ความรู้ของผู้ใช้จะช่วยให้สามารถอ่านผลลัพธ์ที่แสดงบนจอภาพได้อย่างเข้าใจ และพร้อมที่จะกระทำการติดต่อกับระบบได้

2) ภาษาการกระทำ (Action Language) นอกจากผู้ใช้จะมีองค์ความรู้เพียงอย่างเดียวคงไม่เพียงพอต่อการติดต่อกับระบบคอมพิวเตอร์ ที่จะทำให้คอมพิวเตอร์ทราบได้ว่าผู้ใช้ได้มีการกระทำได้โต้ตอบกลับไปบ้าง ซึ่งการกระทำดังกล่าวจะต้องเป็นการกระทำที่ระบบคอมพิวเตอร์เข้าใจ ดังนั้นจึงต้องมี "ภาษาแสดงการกระทำ" ซึ่งหมายถึงภาษาที่ใช้สร้างรูปแบบการส่งผ่านคำสั่งที่จะนำเข้าไปประมวลผลในระบบคอมพิวเตอร์ หรือกล่าวได้ง่าย ๆ คือ "รูปแบบของลักษณะการสื่อสารกับผู้ใช้ (User Interface Mode/Style)" นั่นเอง เช่น ต้องการแสดงการกระทำโต้ตอบกลับไปยังระบบคอมพิวเตอร์ด้วยการเลือกเมนูคำสั่ง การถาม-ตอบ การป้อนข้อมูล เป็นต้น

3) ปฏิกิริยาโต้ตอบของผู้ใช้ (User's Reaction) เป็นปฏิกิริยาที่เกิดขึ้นหลังจากผู้ใช้ได้รับผลลัพธ์จากระบบ ปฏิกิริยาโต้ตอบของผู้ใช้จะนำเข้าสู่ระบบคอมพิวเตอร์อีกครั้งเพื่อประมวลผลเพิ่มเติมตามภาษาที่ทำให้เกิดการกระทำต่อไป

2. กลุ่มของระบบคอมพิวเตอร์

1) บทโต้ตอบ (Dialog) หรือ "บทสนทนา" ที่ใช้ในการติดต่อและสื่อสารกับผู้ใช้ระบบที่คอมพิวเตอร์ได้เตรียมไว้ ซึ่งอาจเป็นชุดข้อมูลหรือชุดคำสั่งก็ได้

2) การประมวลผลของคอมพิวเตอร์ (Computer Processing) จะทำหน้าที่แปลคำสั่งที่ผู้ใช้ป้อนเข้ามาทาง Action Language แล้วปฏิบัติตามคำสั่ง จากนั้นก็สร้างผลลัพธ์ขึ้นเพื่อนำไปเสนอต่อไป โดยเมื่อได้รับคำสั่งจาก Action Language ระบบคอมพิวเตอร์ก็จะเริ่มต้นดำเนินการแปลคำสั่งที่ได้รับเข้ามา จากนั้นก็จะประมวลผลตามคำสั่งเพื่อสร้างเป็นผลลัพธ์แสดงออกทางจอภาพต่อไป

3) ภาษาการนำเสนอ (Presentation Language) เป็นภาษาที่ระบบคอมพิวเตอร์ใช้เพื่อแสดงผลลัพธ์ให้ผู้ใช้ระบบได้รับทราบ กล่าวง่าย ๆ คือรูปแบบการแสดงผลลัพธ์นั่นเอง เช่น การแสดงผลลัพธ์เป็นแบบข้อความ รูปภาพ กราฟ ภาพนิ่ง ภาพเคลื่อนไหว ภาพสามมิติ แสง สี เสียง เป็นต้น โดยการนำเสนอนี้จะเกิดขึ้นเมื่อผ่านการประมวลผลของคอมพิวเตอร์แล้ว และนำเสนอไปตามบทสนทนาหรือบทโต้ตอบ เพื่อให้เกิดการกระทำกลับคืนมาจากผู้ใช้ระบบ

ระบบจัดการสื่อประสานกับผู้ใช้ (User Interface Management System : UIMS)

ระบบจัดการสื่อประสานกับผู้ใช้ เป็นซอฟต์แวร์ที่มีบทบาทเป็นตัวกลางในการจัดการสื่อประสานที่จะนำเสนอผลลัพธ์ที่ได้จากการประมวลผลของระบบสนับสนุนการตัดสินใจไปยังผู้ใช้ในรูปแบบต่าง ๆ ที่เหมาะสม โดยระบบจัดการสื่อประสานจะติดต่อกับระบบจัดการฐานข้อมูล ระบบจัดการฐานแบบจำลอง และระบบจัดการฐานองค์ความรู้ เพื่อนำข้อมูลที่ประมวลผลได้ตามคำสั่งของผู้ใช้มาแสดงผล นอกจากนี้ หากเป็นระบบสนับสนุนการตัดสินใจที่มีขนาดใหญ่ ระบบจัดการสื่อประสานกับผู้ใช้ยังจะต้องสามารถอำนวยความสะดวกให้กับผู้ใช้ในการสร้างหรือเลือกรูปแบบในการแสดงผลเองได้อีกด้วย (โอภาส เอี่ยมสิริวงศ์, 2556)

ความสามารถที่หลากหลายเพื่อให้สามารถตอบสนองความต้องการและประสิทธิภาพของผู้ใช้งาน โดยความสามารถที่สำคัญอาจประกอบด้วยสิ่งต่อไปนี้

1. สามารถรองรับเทคโนโลยีด้านกราฟิกของสื่อประสานกับผู้ใช้ (Graphic User Interface) เพื่อช่วยให้ผู้ใช้สามารถใช้งานระบบสนับสนุนการตัดสินใจได้อย่างง่ายและสะดวกยิ่งขึ้น
2. สามารถรองรับอุปกรณ์นำเข้าข้อมูล (Input Device) ได้หลายประเภท เพื่อช่วยให้สามารถนำข้อมูลจากแหล่งข้อมูลต่าง ๆ เข้าสู่ระบบสนับสนุนการตัดสินใจได้
3. สามารถรองรับการแสดงผลและการใช้งานอุปกรณ์แสดงผลข้อมูลต่าง ๆ ได้
4. มีความยืดหยุ่นและสามารถปรับระบบตามลักษณะปัญหาต่าง ๆ ได้อย่างรวดเร็ว
5. ช่วยจัดเก็บข้อมูลที่ผ่านเข้า-ออกจากระบบสนับสนุนการตัดสินใจได้อย่างมีประสิทธิภาพ
6. สามารถรองรับการแสดงผลในลักษณะภาพกราฟิก 3 มิติ ที่มีความสวยงามได้
7. สามารถแสดงข้อมูลพร้อมกันหลายหน้าต่างและหลายรูปแบบได้
8. สามารถสื่อความต้องการของผู้ใช้ไปยังผู้สร้างระบบสนับสนุนการตัดสินใจผ่านทางระบบสื่อประสานกับผู้ใช้
9. จัดเตรียมตัวอย่างให้ผู้ใช้ได้ฝึกทักษะการใช้ระบบ โดยอาจแนะนำผ่านกระบวนการป้อนข้อมูลเข้าและกระบวนการสร้างแบบจำลอง
10. ผู้ใช้สามารถปรับแต่งหน้าตาของระบบสื่อประสานกับผู้ใช้ให้เป็นไปตามความต้องการของผู้ใช้ได้เสมอ

การออกแบบส่วนการติดต่อผู้ใช้งาน (User InterfaceDesign) เป็นส่วนที่มีความสำคัญอย่างยิ่งในการออกแบบและพัฒนาโปรแกรมประยุกต์ นักพัฒนาจำเป็นต้องมีความรู้ความเข้าใจในหลาย ๆ ด้านเพื่อที่จะสร้างส่วนการติดต่อผู้ใช้ที่สามารถใช้งานร่วมกับโปรแกรมได้อย่างมีประสิทธิภาพมากที่สุด องค์ประกอบที่สำคัญที่จะต้องพิจารณาครอบคลุมด้านการออกแบบส่วนการติดต่อผู้ใช้งานประกอบด้วย ดังต่อไปนี้

1) องค์ความรู้ด้านคอมพิวเตอร์ (Computer Science) ความรู้ด้านคอมพิวเตอร์เกี่ยวข้องกับการนำความรู้ทางด้านเทคโนโลยีสารสนเทศมาประยุกต์ให้เหมาะสมกับการออกแบบและพัฒนาโปรแกรม ตัวอย่างเช่น การเลือกใช้เครื่องมือในการพัฒนาหรือการออกแบบ การเลือกโปรแกรมที่ช่วยในการพัฒนา หรือการประยุกต์ใช้เครื่องมือช่วยในการจัดการวางแผนการพัฒนาโปรแกรม เพื่อให้การพัฒนาเป็นไปอย่างเป็นระบบและมีประสิทธิภาพ

2) องค์ความรู้ด้านจิตวิทยา (Psychology) ความรู้ด้านจิตวิทยาเกี่ยวข้องกับการเข้าใจพฤติกรรมและความคิดของผู้ใช้งาน โดยที่การเข้าใจพฤติกรรมการใช้งานจะส่งผลต่อประสิทธิภาพการใช้งานของผู้ใช้ ส่วนการรับรู้และการสื่อสารระหว่างโปรแกรมและผู้ใช้ผ่านส่วนการติดต่อผู้ใช้เป็นสิ่งสำคัญที่ต้องพิจารณา

3) องค์ความรู้ด้านการยศาสตร์ (Ergonomics) ความรู้ด้านการยศาสตร์เกี่ยวข้องกับการปรับเปลี่ยนสภาพงานให้เหมาะสมกับผู้ใช้หรือการปรับปรุงสภาพการทำงานอย่างเป็นระบบ โดยต้องพิจารณาให้มีความสอดคล้องกับลักษณะการใช้งานและข้อจำกัดทางด้านสรีรศาสตร์ของผู้ใช้ ตัวอย่างเช่น การออกแบบหน้าจอของโปรแกรมที่มีหน้าต่างการทำงานหลาย ๆ งาน จึงต้องพิจารณาว่าขนาดของหน้าต่างที่จะสร้างนั้นจะมีขนาดเล็กที่สุดขนาดใดเพราะเนื่องจากสายตาของผู้ใช้งานแต่ละช่วงอายุมีข้อจำกัดต่างกัน

4) องค์ความรู้ด้านการออกแบบและกราฟิก (Graphic Design) ความรู้ด้านการออกแบบและกราฟิกเกี่ยวข้องกับการออกแบบกราฟิกซึ่งประกอบด้วยกลุ่มของข้อความ รูปภาพ หรือแอนิเมชัน นักออกแบบจะต้องมีองค์ความรู้ทางด้านกราฟิกและทรัพยากรอื่น ๆ เพื่อจัดการวางองค์ประกอบหรือส่วนต่าง ๆ ลงไปในโปรแกรมเพื่อเป็นตัวอย่างในการติดต่อระหว่างผู้ใช้และคอมพิวเตอร์

5) องค์ความรู้ด้านภาษาและการสื่อสาร (Language and Communications) การนำความรู้ด้านภาษาและการสื่อสารมาประยุกต์ในการออกแบบส่วนการติดต่อผู้ใช้งานเป็นสิ่งสำคัญเพื่อให้ผู้ใช้สามารถเข้าใจและสื่อสารกับระบบได้อย่างถูกต้องและสมบูรณ์ โดยในการออกแบบส่วนการติดต่อผู้ใช้งานควรพิจารณาใช้ภาษาที่เข้าใจง่ายและสื่อความหมายได้ถูกต้องตามฟังก์ชันการทำงานของโปรแกรม เพื่อให้ผู้ใช้สามารถใช้งานได้ง่ายและมีประสิทธิภาพ เช่น การใช้ไอคอนที่มีข้อความ “Ext.” แทนคำว่า “Exit” เพื่อสื่อความหมายว่า “ออก” ซึ่งผู้ใช้ที่มีความเข้าใจและสื่อสารกับระบบจะเข้าใจไอคอนดังกล่าวได้ ทำให้การใช้งานเป็นไปได้อย่างรวดเร็วและสะดวกสบาย

การนำเอาความรู้ด้านเทคโนโลยีสารสนเทศคอมพิวเตอร์ จิตวิทยา กายศาสตร์และการออกแบบและกราฟิกมารวมกันในการออกแบบส่วนการติดต่อผู้ใช้งานจะช่วยให้สามารถสร้างประสบการณ์การใช้งานที่มีประสิทธิภาพและเข้ากันได้กับผู้ใช้ นอกจากนี้ยังมีการประยุกต์ใช้กราฟิกให้มีความสวยงามและน่าสนใจ เพื่อสร้างความพึงพอใจและความสนุกสนานในการใช้งาน

ประเภทของส่วนติดต่อผู้ใช้

ส่วนการติดต่อผู้ใช้งานหรือ User Interface (UI) แบ่งออกเป็น 2 ประเภทหลัก ๆ ดังนี้

1. Command-Line Interface (CLI)

CLI เป็นรูปแบบการติดต่อผู้ใช้งานที่ใช้ข้อความเป็นตัวกลางในการสื่อสารระหว่างผู้ใช้งานและระบบคอมพิวเตอร์ ผู้ใช้งานจะต้องป้อนคำสั่งหรือคำสั่งของโปรแกรมเพื่อทำงานที่ต้องการ ตัวอย่าง CLI ที่รู้จักกันดีคือหน้าต่างคำสั่ง (Command Prompt) ในระบบปฏิบัติการ Windows หรือเทอร์มินัล (Terminal) ในระบบปฏิบัติการ Linux/Unix การใช้งาน CLI ต้องการความรู้เกี่ยวกับคำสั่งและพารามิเตอร์ที่เกี่ยวข้อง และมักเน้นไปที่ผู้ใช้งานที่มีความเชี่ยวชาญในการใช้งานระบบคอมพิวเตอร์มากกว่าผู้ใช้ทั่วไป

2. Graphical User Interface (GUI)

GUI เป็นรูปแบบการติดต่อผู้ใช้งานที่ใช้กราฟิกเป็นตัวกลางในการสื่อสารระหว่างผู้ใช้งานและระบบคอมพิวเตอร์ ใน GUI ผู้ใช้งานสามารถจัดการกับระบบได้ผ่านหน้าต่างที่มีสัญลักษณ์ (Icon) และองค์ประกอบอื่น ๆ เช่น ปุ่ม (Button) แถบเมนู (Menu bar) และช่องข้อมูล (Text box) ผู้ใช้งานสามารถดำเนินการโดยการคลิกเลือก ลาก หรือป้อนข้อมูลผ่านองค์ประกอบต่าง ๆ ที่ให้ไว้ใน UI โดยทั่วไป GUI เป็นส่วนการติดต่อที่ใช้งานง่ายและสื่อสารได้อย่างชัดเจน จึงเป็นที่นิยมในโปรแกรมและแอปพลิเคชันในปัจจุบัน

สรุปแล้ว GUI และ CLI เป็นรูปแบบส่วนการติดต่อผู้ใช้งานที่มีความแตกต่างกัน โดย CLI ใช้ข้อความเป็นตัวกลางในการสื่อสารและต้องการความรู้เพิ่มเติมในการใช้งาน ในขณะที่ GUI ใช้กราฟิกเป็นตัวกลางในการสื่อสารและเน้นความสะดวกสบายในการใช้งานสำหรับผู้ใช้ทั่วไป

หลักการออกแบบส่วนติดต่อผู้ใช้

หลักการออกแบบของจาคอป (Jakob's Principle of Design) เป็นแนวคิดที่ใช้ในการออกแบบส่วนการติดต่อผู้ใช้สำหรับระบบหรือโปรแกรมที่นำมาใช้งาน แนวทางดังกล่าวได้รับความนิยมและใช้ในอุตสาหกรรมซอฟต์แวร์เป็นที่ยอมรับ ได้สรุปสาระสำคัญของการออกแบบให้ใช้งานง่ายไว้ 10 ข้อ (Yablonski, J., 2020) ดังต่อไปนี้

1. การมองเห็นสถานะระบบ (Visibility of system status)

เป็นหลักการสำคัญในการออกแบบส่วนการติดต่อผู้ใช้ที่มุ่งเน้นให้ผู้ใช้งานสามารถรับรู้สถานะปัจจุบันของระบบได้อย่างชัดเจน ซึ่งระบบควรแสดงข้อมูลหรือสัญญาณให้ผู้ใช้งานทราบว่าระบบกำลังดำเนินการอะไรอยู่ในขณะนั้น และมีอะไรเกิดขึ้นเมื่อระบบตอบสนองต่อคำสั่งหรือกระทำของผู้ใช้งาน การแสดงสถานะระบบได้อย่างชัดเจนช่วยให้ผู้ใช้งานรับรู้ว่าระบบกำลังทำงาน หรือกำลังดำเนินกระบวนการใด ๆ เพื่อให้ผู้ใช้งานทราบถึงผลลัพธ์หรือสถานะปัจจุบันของระบบได้ การตอบสนองจากระบบควรเป็นไปในระยะเวลาที่เหมาะสมต่อการโต้ตอบจากผู้ใช้งาน ไม่ควรช้าหรือเร็วเกินไป เพื่อให้ผู้ใช้งานรับรู้ผลลัพธ์และสามารถปรับตัวหรือดำเนินการต่อได้ตรงกับที่ต้องการ

2. ใช้ภาษาเดียวกัน (Match between system and the real world)

เป็นหลักการสำคัญในการออกแบบส่วนการติดต่อผู้ใช้ที่มุ่งเน้นให้การใช้งานเป็นไปในแบบที่เข้าใจง่ายและสอดคล้องกับโลกจริงที่ผู้ใช้เคยพบเห็นและรู้จัก หลักการนี้นำเสนอว่าการออกแบบควรใช้ภาษาและสัญลักษณ์ที่คล้ายคลึงกับคำศัพท์และแบบสัญลักษณ์ที่ผู้ใช้รู้จักในชีวิตประจำวัน เพื่อให้ผู้ใช้เข้าใจและสื่อสารได้อย่างรวดเร็วและมีประสิทธิภาพ ในการออกแบบส่วนการติดต่อผู้ใช้งานควรใช้ภาษาที่เข้าใจง่ายและเหมือนกับวิธีการสื่อสารที่ผู้ใช้เคยใช้ในชีวิตประจำวัน ตัวอย่างเช่น การใช้คำสั่งที่คล้ายคลึงกับคำสั่งที่เราใช้ในการพูดหรือใช้คำสั่งที่แน่นอนและตรงกับสิ่งที่ผู้ใช้คาดหวัง การใช้ภาษาและสัญลักษณ์ที่เข้าใจง่ายและเป็นที่รู้จัก เช่น การใช้ไอคอนที่คล้ายคลึงกับวัตถุจริง หรือการแสดงผลข้อความในรูปแบบที่เข้าใจง่าย

3. อิสระจากการควบคุม (User control and freedom)

เป็นหลักการสำคัญในการออกแบบส่วนการติดต่อผู้ใช้ที่มีเป้าหมายในการเพิ่มความสะดวกสบายให้กับผู้ใช้งานและป้องกันปัญหาที่อาจเกิดจากความผิดพลาดของผู้ใช้ หลักการนี้เน้นให้ผู้ใช้งานมีความควบคุมและอิสระในการทำงานกับระบบ โดยระบบควรทำให้ผู้ใช้งานสามารถกลับไปสู่สถานะเดิมหรือย้อนกลับไปในกรณีที่ทำผิดพลาด นอกจากนี้ระบบควรมีการให้ผู้ใช้งานสามารถแก้ไขหรือยกเลิกการกระทำที่เกิดขึ้น (Undo/Redo) ซึ่งเป็นฟังก์ชันพื้นฐานที่ควรมีในโปรแกรมทุก ๆ โปรแกรม การให้ความควบคุมและอิสระให้กับผู้ใช้งานช่วยลดความเครียดและความไม่สมดุลของผู้ใช้งาน และเพิ่มประสิทธิภาพในการทำงาน ผู้ใช้งานจะรู้สึกมั่นใจและมีความ

เชื่อมั่นในการใช้งานระบบ เนื่องจากพวกเขามีความควบคุมและอิสระในการจัดการกับระบบได้อย่างสมบูรณ์ หลักการนี้เป็นส่วนสำคัญในการออกแบบโปรแกรมและระบบที่ผู้ใช้งานต้องใช้งานอยู่ทุกวันนี้ และมีผลในการสร้างประสบการณ์การใช้งานที่ดีและเป็นที่ยอมรับ

4. ความคงสภาพและความมีมาตรฐาน (Consistency and standards)

การออกแบบส่วนการติดต่อผู้ใช้ที่มุ่งเน้นให้รูปแบบและการทำงานของระบบเป็นไปในลักษณะที่เหมือนกันทั้งระบบ ความสอดคล้องกันนี้ช่วยให้ผู้ใช้งานสามารถเข้าใจและใช้งานได้ง่ายขึ้น เมื่อมีความสอดคล้องกันในระบบ ผู้ใช้งานสามารถใช้งานส่วนต่าง ๆ ของระบบได้อย่างเร็วและมีประสิทธิภาพ เนื่องจากพวกเขาสามารถนำความรู้และประสบการณ์ที่ได้รับจากการใช้งานส่วนอื่น ๆ ในระบบมาใช้ได้ ความคงสภาพและความมีมาตรฐานเป็นที่สำคัญในการออกแบบเนื่องจากมีผลต่อประสบการณ์ของผู้ใช้งาน ความสอดคล้องกันที่ชัดเจนในระบบช่วยลดความสับสนและความผิดพลาดที่อาจเกิดขึ้นจากความไม่เข้าใจในการใช้งานระบบ นอกจากนี้ยังช่วยให้ผู้ใช้งานสามารถนำประสบการณ์ที่ได้รับจากการใช้งานส่วนต่าง ๆ ในระบบมาใช้ได้อย่างต่อเนื่อง โดยไม่ต้องเรียนรู้ใหม่ในทุกครั้งที่เข้าถึงส่วนใหม่

5. การป้องกันความผิดพลาด (Error prevention)

การจัดเตรียมเครื่องมือและวิธีการที่ช่วยลดความเป็นไปได้ของความผิดพลาดที่ผู้ใช้งานอาจทำในระบบหรือโปรแกรมที่ออกแบบไว้ หากเกิดความผิดพลาดในการป้อนข้อมูลหรือการกระทำอื่น ๆ ที่ไม่ถูกต้อง อาจส่งผลให้เกิดปัญหาหรือความเสียหายต่อระบบหรือผู้ใช้งานได้ ในการป้องกันความผิดพลาด การออกแบบควรจัดเตรียมเครื่องมือและวิธีการที่ช่วยให้ผู้ใช้งานทำงานได้อย่างถูกต้องและมีประสิทธิภาพ เช่นการตรวจสอบข้อมูลที่ผู้ใช้ป้อนเข้าระบบ เพื่อตรวจสอบความถูกต้องและความสมบูรณ์ การใช้ข้อความแจ้งเตือนเมื่อเกิดข้อผิดพลาดเพื่อแจ้งให้ผู้ใช้งานทราบเกี่ยวกับปัญหาที่เกิดขึ้น และการออกแบบวิธีใช้งานที่ชัดเจนและเข้าใจง่าย เพื่อลดความผิดพลาดที่อาจเกิดขึ้นจากความสับสนหรือไม่เข้าใจในวิธีการทำงานของระบบ

6. การรับรู้มากกว่าการเรียกคืน (Recognition rather than recall)

เป็นหลักการที่มุ่งเน้นให้การออกแบบให้ระบบต้องมีการจัดการเพื่อไม่ให้ผู้ใช้ต้องจำคำสั่งต่าง ๆ ในการใช้งานโปรแกรม เพราะหากโปรแกรมต้องบังคับให้ผู้ใช้ต้องจำคำสั่งมากเกินไป ส่งผลต่อประสิทธิภาพการใช้งานและความรวดเร็วในการใช้งานโปรแกรมด้วย ด้วยปัญหานี้ โปรแกรมสมัยก่อนที่มีการประมวลผลและการป้อนกลับให้ผู้ใช้แบบ รายบรรทัด (Command-Line Interface) ซึ่งส่งผลให้ผู้ใช้ ใช้งานได้ไม่สะดวก และการใช้ยังของโปรแกรมยังจำกัดอยู่ในวงแคบเฉพาะกลุ่มคนเท่านั้น จึงนำไปสู่การออกแบบโปรแกรมที่ใช้งานง่ายกว่า เพียงการคลิกและแนวทางการพัฒนาที่เน้นส่วนการติดต่อของผู้ใช้ อย่าง Graphical User Interface (GUI) ทุกวันนี้

7. ประสิทธิภาพและความยืดหยุ่นในการใช้งาน (Flexibility and efficiency of use)

เป็นหลักการที่มุ่งเน้นให้การออกแบบให้ระบบต้องมีความยืดหยุ่นของการใช้งาน และมุ่งเน้นให้โปรแกรมทำงานและตอบสนองผู้ใช้มากที่สุด นักพัฒนาจะต้องออกแบบให้ระบบสามารถใช้งานได้กับคนในองค์กรได้หลาย ๆ แผนก หมายความว่า แผนกการเงิน ก็สามารถเข้าไปใช้งานและรู้หลักการทำงานของโปรแกรมของแผนกอื่น ๆ เข้าใจฟังก์ชั่นการทำงาน เป็นต้น

8. การออกแบบสวยงามและเรียบง่าย (Aesthetic and minimalist design)

เป็นหลักการออกแบบมีความง่ายต่อการใช้งานและมีความสวยงาม โดยผู้ใช้งานสามารถเข้าใจและควบคุมระบบได้โดยใช้อุปกรณ์นำเข้าได้ เช่น การใช้เมาส์คลิก หรือใช้คีย์บอร์ดในการป้อนข้อมูล เป็นต้น

9. การช่วยเหลือผู้ใช้ (Help users from errors)

เป็นหลักการที่มุ่งเน้นให้การออกแบบระบบ ให้ความสำคัญกับการแสดงเตือน เมื่อเกิดความผิดพลาดใด ๆ ขึ้นในระบบ โดยการแสดงข้อความใด ๆ ควรจะแสดงในรูปแบบที่เข้าใจง่ายพร้อมทั้งแสดงรายละเอียดวิธีการแก้ไขปัญหา พร้อมทั้งการอธิบายวิธีการต่าง ๆ อย่างชัดเจน โดยทั้งหมดนี้ผู้ใช้จะต้องสามารถแก้ไขปัญหาที่เกิดขึ้นได้เอง

10. การสร้างเอกสารและคู่มือ (Help and documentation)

เป็นหลักการที่มุ่งเน้นให้การออกแบบระบบนั้นจะต้องมีคู่มือที่ดีประกอบการใช้งาน แม้นักพัฒนาจะคาดหวังให้ผู้ใช้อาศัยการใช้งานคู่มือให้น้อยที่สุดก็ตาม คู่มือที่ดีควรมีการอธิบายรายละเอียดและขั้นตอนการใช้งานอย่างชัดเจน มีตัวอย่างหรือภาพประกอบที่ช่วยในการเข้าใจ และมีการจัดเรียงลำดับขั้นตอนการแก้ไขปัญหาที่เป็นไปได้

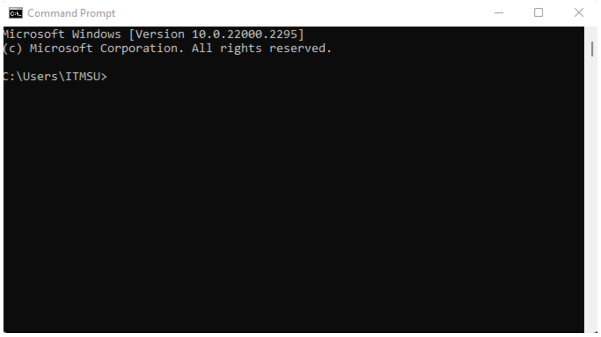
หลักการออกแบบของจาคอปเป็นเครื่องมือที่มีประสิทธิภาพในการออกแบบส่วนการติดต่อผู้ใช้ และใช้กันอย่างแพร่หลายในอุตสาหกรรมซอฟต์แวร์ การปฏิบัติตามหลักการดังกล่าวช่วยให้ผู้ใช้สามารถใช้งานและเข้าใจระบบได้อย่างมีประสิทธิภาพและเป็นเสริมสร้างประสบการณ์การใช้งานที่ดีและเชื่อถือได้

การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface)

เป็นการออกแบบจอภาพเพื่อให้ผู้ใช้งานสามารถโต้ตอบกับระบบใด ๆ ตามความต้องการอย่างมีประสิทธิภาพ ในปัจจุบันนิยมใช้การออกแบบส่วนติดต่อกับผู้ใช้แบบกราฟิก (Graphic User Interfaceซึ่งสามารถสื่อสารกับผู้ใช้ในรูปแบบข้อความและรูปภาพต่าง ๆ ทำให้การใช้งานง่ายและเรียนรู้ได้รวดเร็ว สำหรับรูปแบบการออกแบบส่วนติดต่อกับผู้ใช้ (โอภาส เอี่ยมสิริวงศ์, 2556) มีดังต่อไปนี้

1. การโต้ตอบด้วยคำสั่ง (Command Language Interaction)

เป็นการโต้ตอบกับระบบโดยที่ผู้ใช้จะต้องพิมพ์คำสั่งลงในช่องป้อนคำสั่ง เพื่อกระตุ้นให้เกิดการทำงานในระบบ จึงเหมาะสำหรับผู้ชายที่มีความชำนาญในการใช้ระบบ ดังภาพที่ 6.2



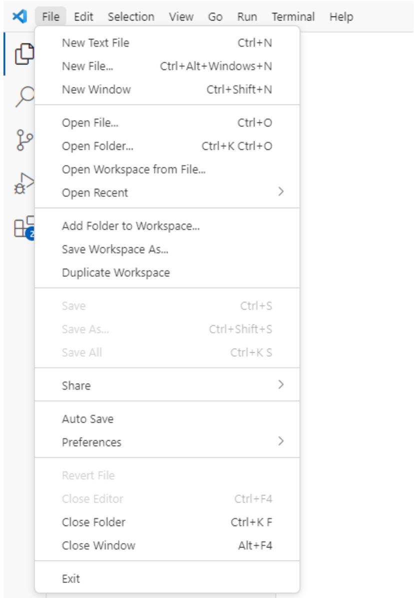
ภาพที่ 6.2 หน้าจอการโต้ตอบด้วยคำสั่ง

การโต้ตอบกับระบบประเภทนี้ต้องการผู้ใช้ระบบที่มีความชำนาญในการใช้ระบบปฏิบัติการ DOS และความเข้าใจในวิธีการป้อนคำสั่งและกฎการใช้งานของระบบเฉพาะเจาะจงมากมาย แต่ในระบบงานธุรกิจปัจจุบันมีผู้ใช้ระบบที่ไม่มีความชำนาญในการป้อนคำสั่งประเภทนี้มากมาย และการป้อนคำสั่งไม่เหมาะสมกับการทำงานด้านธุรกิจ เช่น งานด้านเอกสารธุรกิจ งานด้านการคำนวณ ดังนั้นการโต้ตอบกับระบบด้วยคำสั่งจึงไม่ได้รับความนิยมในการนำมาใช้กับระบบงานธุรกิจโดยทั่วไป และในปัจจุบันหน้าจอแสดง Graphic User Interface (GUI) ได้รับความนิยม ทำให้การโต้ตอบกับระบบด้วยการป้อนคำสั่งประเภทนี้ได้รับความนิยมลดลง

2. การโต้ตอบด้วยเมนูคำสั่ง (Menu Interaction)

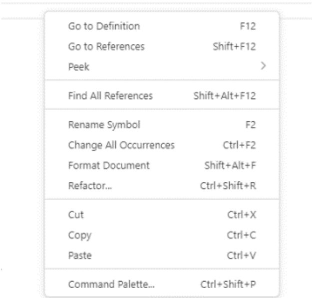
เป็นการโต้ตอบกับระบบด้วยการแสดงเมนูคำสั่งให้ผู้ใช้เลือกคำสั่งใด ๆ เพื่อติดต่อกับระบบโดยผู้ใช้ไม่ต้องป้อนคำสั่งเอง ดังนั้นจึงได้รับความนิยม ความง่ายในการใช้งานและการทำความเข้าใจการพัฒนาการออกแบบส่วนการติดต่อกับผู้ใช้ในปัจจุบันได้มีการนำเสนอเมนูคำสั่งแบบกราฟิก (GUI Menu) ซึ่งเป็นการพัฒนาโดยใช้ภาษาโปรแกรมระดับสูงในยุคที่ 4 (4GLs) และภาษาเชิงวัตถุ (Object-oriented Language) ซึ่งสามารถทำงานบนพื้นฐานระบบปฏิบัติการ Windows หรือ macOS ได้ รูปแบบของเมนูดังกล่าวมีลักษณะ ดังนี้

2.1 Pull-down Menu เป็นเมนูที่แสดงตัวเลือกของรายการคำสั่ง โดยรายการคำสั่งจะปรากฏทันทีที่ผู้ใช้เลือกจากแถบเมนู ซึ่งตัวเลือกนั้นจะเรียงลงมาจากบนลงล่างใน Pull-down Menu โดยแต่ละกลุ่มของรายการเมนูคำสั่งจะถูกจัดให้รวมอยู่ในกลุ่มเดียวกัน เช่น New, Open, และ Close จะถูกจัดให้รวมอยู่ในกลุ่ม File และแสดงตาม Pull-down Menu ดังภาพที่ 6.3



ภาพที่ 6.3 เมนูแสดง Pull-down Menu

2.2 Pop-up Menu เป็นเมนูคำสั่งอีกชนิดหนึ่งที่แสดงรายการคำสั่ง โดยที่ผู้ใช้เป็นผู้ทำให้เกิดเมนูชนิดนี้ขึ้น โดยการนำเมาส์ไปวางที่ข้อความ ออบเจ็กต์หรือบริเวณใดก็ได้ที่ผู้ใช้สนใจในจอภาพ จากนั้นคลิกเมาส์ขวาจะปรากฏ Pop-up Menu ซึ่งภายในจะประกอบไปด้วยคำสั่งและคุณสมบัติที่เกี่ยวกับข้อความหรือออบเจ็กต์ที่ผู้ใช้สนใจในการทำงาน เช่น Pop-up Menu ที่ปรากฏเมื่อคลิกขวาบนไอคอนในระบบปฏิบัติการ ดังภาพที่ 6.4



ภาพที่ 6.4 เมนูแสดง Pop-up Menu

ออกแบบระบบแบบเมนูคำสั่งให้มีประสิทธิภาพและสามารถสื่อสารกับผู้ใช้เพื่อการทำงานได้อย่างมีประสิทธิภาพ ควรมีหลักเกณฑ์ในการออกแบบเมนูคำสั่งที่ถูกต้อง ดังนี้

1) แต่ละเมนูคำสั่งควรเลือกใช้คำที่สามารถสื่อความหมายได้อย่างชัดเจน คำสั่งควรถูกเลือกให้มีความหมายที่ชัดเจนและเข้าใจง่ายเมื่อผู้ใช้มองเห็นเมนูคำสั่งนั้น ๆ

2) ควรมีการใช้ตัวอักษรพิมพ์ใหญ่หรือตัวพิมพ์เล็กตามความเหมาะสม การใช้ตัวอักษรพิมพ์ใหญ่หรือตัวพิมพ์เล็กควรเป็นไปตามเกณฑ์และรูปแบบการออกแบบที่เหมาะสมกับผู้ใช้งานและรูปแบบเมนู

3) ควรมีการจัดกลุ่มคำสั่งที่มีความเกี่ยวข้องกันไว้ในกลุ่มเดียวกัน คำสั่งที่มีความสัมพันธ์กันควรถูกจัดเข้าไปในกลุ่มเมนูเดียว เช่น คำสั่งที่เกี่ยวกับไฟล์จะถูกจัดในกลุ่มเมนู "File"

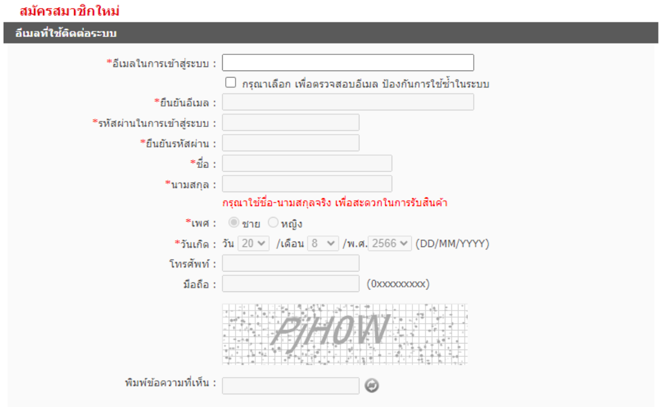
4) ไม่ควรมีจำนวนเมนูคำสั่งมากเกินไป ควรจำกัดจำนวนเมนูคำสั่งให้เหมาะสมและไม่มากเกินไป เพื่อป้องกันความซับซ้อนและความยุ่งยากในการใช้งาน

5) ควรมีเมนูย่อยสำหรับเมนูคำสั่งที่มีการทำงานย่อยภายในไม่มากเกินไป ควรพิจารณาการใช้เมนูย่อยเพื่อลดความซับซ้อนและเพิ่มความชัดเจนในการใช้งาน

6) เมื่อมีการเลือกเมนูคำสั่ง ควรออกแบบให้มีแถบสีปรากฏที่เมนูคำสั่งที่ถูกเลือก เพื่อแสดงให้ผู้ใช้รู้ว่าเมนูคำสั่งได้รับการเลือกและกำลังใช้งานอยู่ ควรออกแบบการแสดงผลที่ชัดเจนเมื่อมีการเลือกเมนูคำสั่ง

3. การโต้ตอบด้วยแบบฟอร์ม (Form Interaction)

เป็นการโต้ตอบที่ผู้ใช้ระบบจะต้องป้อนข้อมูลลงในช่องว่างที่อยู่ ในแบบฟอร์มที่แสดงทางหน้าจอ ซึ่งมีลักษณะคล้ายกับคำร้องในกระดาษ การออกแบบฟอร์มที่ดีควรจะมีชื่อของช่องป้อนข้อมูลที่สื่อความหมายชัดเจน มีการแบ่งส่วนของข้อมูลบนแบบฟอร์ม และช่องป้อนข้อมูลของแบบฟอร์มไม่ควรมีความยาวมากเกินไป ดังภาพที่ 6.5


สมัครสมาชิกใหม่

อีเมลที่ใช้ติดต่อระบบ

*อีเมลในการเข้าสู่ระบบ :

กรุณาเลือก เพื่อตรวจสอบอีเมล ป้องกันการใช้ซ้ำในระบบ

*ยืนยันอีเมล :

*รหัสผ่านในการเข้าสู่ระบบ :

*ยืนยันรหัสผ่าน :

*ชื่อ :

*นามสกุล :

กรุณาใช้ชื่อ-นามสกุลจริง เพื่อสะดวกในการรับสินค้า

*เพศ : ชาย หญิง

*วันเกิด : วัน 20 /เดือน 8 /พ.ศ. 2566 (DD/MM/YYYY)

โทรศัพท์ :

มือถือ : (0xxxxxxxxx)

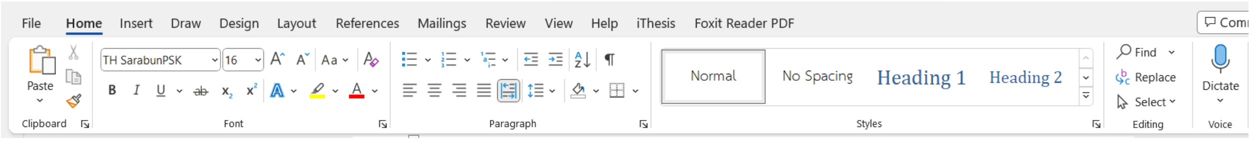
PjHOW

พิมพ์ข้อความที่เห็น :

ภาพที่ 6.5 การโต้ตอบด้วยแบบฟอร์ม

4. การโต้ตอบด้วยการทำงานเชิงวัตถุ (Object-Based Interaction)

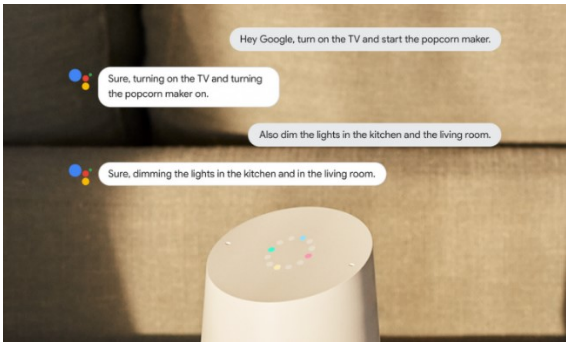
เป็นการโต้ตอบที่ผู้ใช้ระบบที่ใช้สัญลักษณ์ เปลี่ยนตัวแทนคำสั่งที่ใช้ในการปฏิบัติงาน การใช้สัญลักษณ์รูปภาพแทนคำสั่งการทำงานหรือเรียกว่า ไอคอน โดยผู้ใช้สามารถคลิกเมาส์ที่ไอคอนเพื่อสั่งให้โปรแกรมทำงานตามต้องการได้ ข้อดีคือประหยัดพื้นที่บนหน้าจอ ผู้ใช้งานยังสามารถทำความเข้าใจกับคำสั่งที่ถูกแทนด้วยไอคอนได้อย่างง่ายดาย ดังภาพที่ 6.6



ภาพที่ 6.6 การโต้ตอบด้วยการทำงานเชิงวัตถุ

5. การโต้ตอบด้วยภาษามนุษย์ (Natural Language Interaction)

เป็นการโต้ตอบกับระบบด้วยการใช้เสียงพูดของผู้ใช้ระบบ ไม่ว่าจะเป็นการนำเข้าข้อมูลหรือออกจากระบบ ภาษาที่ใช้ เช่น ภาษาอังกฤษ ภาษาไทย เป็นต้น ดังภาพที่ 6.7



ภาพที่ 6.7 การโต้ตอบด้วยภาษามนุษย์

การออกแบบส่วนติดต่อผู้ใช้สมัยใหม่

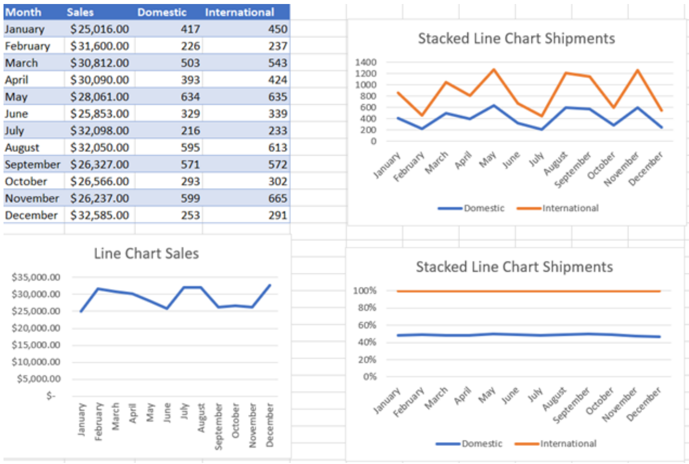
นอกเหนือจากรูปแบบของสื่อประสานกับผู้ใช้ที่ได้กล่าวมาข้างต้นแล้ว เช่น การโต้ตอบด้วยคำสั่ง การโต้ตอบด้วยแบบฟอร์ม และการโต้ตอบด้วยเมนู ยังมีการโต้ตอบแบบอื่น ๆ ที่มีความทันสมัยและสร้างความสมจริงในการใช้งานระบบของผู้ใช้มากขึ้น ตัวอย่างเช่น กราฟิก มัลติมีเดีย และไฮเปอร์มีเดีย ระบบความเป็นจริงเสมือน และการประมวลผลภาษาธรรมชาติ

การใช้รูปแบบอื่น ๆ เหล่านี้ช่วยให้สื่อประสานกับผู้ใช้มีความหลากหลายและมีประสิทธิภาพที่ดีขึ้น โดยรูปแบบเหล่านี้สามารถเพิ่มประสบการณ์การใช้งานที่สมจริงของผู้ใช้และทำให้ระบบดูมีความเป็นจริงมากขึ้น ตัวอย่างเช่น กราฟิกที่ใช้ภาพและภาพเคลื่อนไหว เพื่อแสดงข้อมูลหรือสถานะให้ผู้ใช้เข้าใจได้ง่ายขึ้น มัลติมีเดียที่ให้การสื่อสารแบบเต็มรูปแบบเพื่อให้ผู้ใช้เกิดความต่อเนื่องในการติดตามข้อมูล และไฮเปอร์มีเดียที่สร้างประสบการณ์ที่เสมือนจริงให้กับผู้ใช้

1. กราฟิก (Graphics)

กราฟิกเป็นวิธีหนึ่งในการนำเสนอข้อมูลหรือผลลัพธ์โดยใช้ภาพเป็นสื่อสาร ซึ่งสามารถเป็นรูปภาพเพียงอย่างเดียวหรือรวมกับข้อความเพื่อให้ผู้ใช้สามารถเข้าใจความหมายของข้อมูลและสารสนเทศที่นำเสนอได้อย่างชัดเจนมากยิ่งขึ้น โดยเฉพาะอย่างยิ่งในผลลัพธ์ที่ได้จากระบบสนับสนุนการตัดสินใจ ซึ่งมักจะเป็นข้อมูลจำนวนมากที่ต้องการทำให้ผู้ใช้อ่านข้อมูลเสียเวลาน้อยลง ถ้าแทนการอ่านข้อมูลด้วยภาพจะทำให้ดูง่ายขึ้น เช่น การแสดงข้อมูลด้วยแผนภูมิ (Chart) เป็นต้น

ในปัจจุบันโปรแกรมประมวลกราฟิกมีจำนวนเพิ่มมากขึ้น ไม่ว่าจะเป็นโปรแกรมกราฟิกทางด้านการออกแบบ การสร้างและตกแต่งภาพ ส่วนที่ได้รับความนิยมในปัจจุบันเช่น AutoCAD, Adobe Photoshop, Adobe Illustrator, Freehand, Adobe PageMaker เป็นต้น และสำหรับโปรแกรมที่สามารถจัดการกับข้อมูลให้แสดงผลเป็นแผนภูมิที่ได้รับความนิยมคือ Microsoft Excel ซึ่งตัวอย่างผลลัพธ์ที่ได้แสดงไว้ในภาพที่ 6.8



Month	Sales	Domestic	International
January	$25,016.00	417	450
February	$31,600.00	226	237
March	$30,812.00	503	543
April	$30,090.00	393	424
May	$28,061.00	634	635
June	$25,853.00	329	339
July	$32,098.00	216	233
August	$32,050.00	595	613
September	$26,327.00	571	572
October	$26,566.00	293	302
November	$26,237.00	599	665
December	$32,585.00	253	291

ภาพที่ 6.8 โครงสร้างการทำงานของโปรแกรมกราฟิก

จากภาพที่ 6.8 ที่แสดงการใช้แผนภูมิเพื่อแสดงผลลัพธ์แทนการใช้ข้อมูลแบบข้อความธรรมดา สามารถเห็นความแตกต่างของผลลัพธ์จากแผนภูมิจะช่วยสังเกตเห็นค่าของข้อมูลที่มีค่าต่ำสุดและสูงสุดได้ง่ายและรวดเร็วกว่าการแสดงผลแบบข้อความธรรมดา

ภาพกราฟิกที่สร้างขึ้นสามารถมีรูปแบบหลายรูปแบบ และความสามารถของโปรแกรมประเภทกราฟิกสามารถแสดงผลลัพธ์ได้ในรูปแบบต่างๆ ดังต่อไปนี้

1) Text เป็นการแสดงข้อความที่อธิบายถึงรายละเอียดที่อยู่ในแผนภูมิ เช่น ชื่อแผนภูมิ (Title) ชื่อของชุดข้อมูล (Data Series) ค่าของชุดข้อมูล (Data Value) เป็นต้น

2) Time-series chart สามารถแสดงค่าตัวแปรที่เปลี่ยนไปในทุก ๆ ช่วงเวลาตั้งแต่หนึ่งตัวแปรขึ้นไป

3) Bar and Pie chart สามารถแสดงค่าขององค์ประกอบและค่าผลรวมต่าง ๆ ได้

4) Scatter diagram แสดงความสัมพันธ์ระหว่างค่าตัวแปรตั้งแต่ 2 ตัวแปรขึ้นไป เช่น การแสดงความสัมพันธ์ของจำนวนเจ้าหน้าที่สายการบินที่ปฏิบัติงานในวันจันทร์ และวันอังคาร เป็นต้น

5) Map แผนที่ที่ใช้แสดงความสัมพันธ์ระหว่างข้อมูล 2 กลุ่ม สามารถแสดงได้ทั้ง 2 มิติและ 3 มิติ

6) Layout diagram สำหรับสร้างภาพจำลองสิ่งก่อสร้างต่าง ๆ เช่น ห้องทำงาน หรือศูนย์การค้า เป็นตัวอย่างเพื่อถ่ายทอดสารสนเทศออกมาเป็น Diagram ที่ดูให้เข้าใจได้อย่างง่ายดาย

7) Hierarchy chart สำหรับสร้างแผนผังลำดับชั้น ซึ่งเป็นที่นิยมใช้กันอย่างกว้างขวาง เช่น Organization Chart เป็นต้น

8) Sequence chart เช่น Flowchart เป็นตัวอย่างที่ใช้เรียงลำดับก่อนหลังให้กับกิจกรรมที่เกี่ยวข้อง ซึ่งกิจกรรมในที่นี้เป็นกิจกรรมที่มีความสัมพันธ์เท่าๆ กัน

9) Motion graphic เช่น รูปภาพเคลื่อนไหวในโทรทัศน์ เป็นต้น ที่มีวัตถุประสงค์เพื่อต้องการติดตามดูความเคลื่อนไหวของตัวแปรต่าง ๆ ว่ามีค่าเปลี่ยนแปลงไปหรือไม่อย่างไร

10) Desktop publishing system เป็นระบบที่มีความสามารถทางด้านกราฟิกที่หลากหลาย และเป็นที่นิยมใช้มาก เช่น สามารถถ่ายโอน (Transfer) รูปภาพจากนิตยสารเข้าสู่เครื่องคอมพิวเตอร์โดยใช้เครื่องสแกนเนอร์ ซึ่งอาจสแกนและเก็บภาพที่สแกนไว้ในไดเร็กทอรีใดก็ได้ จากนั้นจึงพิมพ์ภาพออกมา

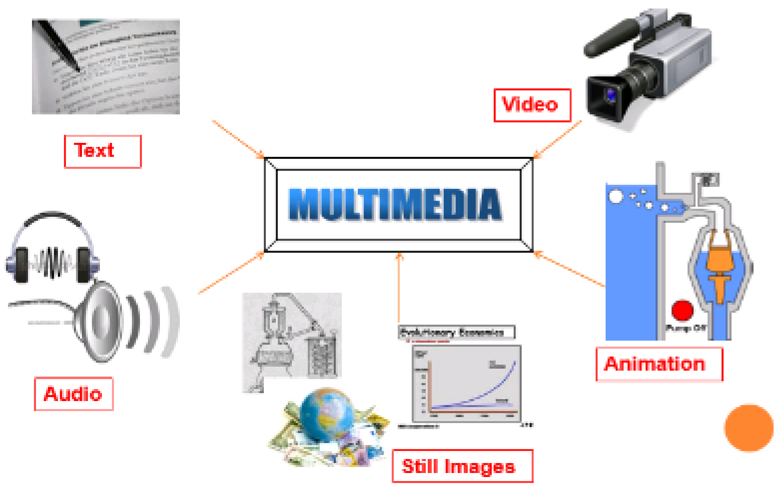
ดังนั้น การใช้ระบบ Computer Graphic และโปรแกรมกราฟิกที่มีหลากหลายรูปแบบสามารถสร้างแผนภูมิและภาพกราฟิกต่าง ๆ ได้เพื่อการนำเสนอข้อมูลที่น่าสนใจและชัดเจนให้กับผู้ใช้งาน

2. มัลติมีเดียและไฮเปอร์เท็กซ์ (Multimedia and Hypertext)

ระบบสนับสนุนการตัดสินใจสามารถตกแต่งลักษณะประสานกับผู้ใช้ให้มีรูปแบบที่น่าใช้งานมากยิ่งขึ้นด้วย "มัลติมีเดีย (Multimedia)" ซึ่งในระบบสนับสนุนการตัดสินใจบ่อยครั้งนั้นจะยิ่งมีแนวโน้มการใช้งานมัลติมีเดียสูงขึ้น การประมวลผลของระบบสนับสนุนการตัดสินใจทุก ๆ ระบบจะต้องใช้อุปกรณ์ที่เกี่ยวกับเทคโนโลยีมัลติมีเดียต่าง ๆ เพื่อการประมวลผลสารสนเทศและการตัดสินใจของผู้บริหารในแต่ละครั้งเพื่อเพิ่มประสิทธิภาพในการตัดสินใจ

มัลติมีเดีย (Multimedia) เป็นการกล่าวถึงประโยชน์โดยรวมของสื่อมากมายที่ทำหน้าที่เป็นลักษณะกลางการสื่อสารระหว่างมนุษย์กับเครื่องจักร โดยใช้กลไกของระบบ (Human-Machine Communication Media) ซึ่งนำเข้าใช้เทคโนโลยีสารสนเทศมากมาย กล่าวคือ เทคโนโลยีสารสนเทศจะใช้คอมพิวเตอร์เพื่อการปรับปรุงการสื่อสารระหว่างมนุษย์กับเครื่องจักร (Human-machine Communication) โดยใช้คุณประโยชน์ของสื่อชนิด

ต่าง ๆ ร่วมกับระบบการประมวลผล ซึ่งทำหน้าที่เป็นหัวใจของการปฏิบัติงานในระบบ DSS ผลลัพธ์ที่ได้อาจแสดงด้วยเสียง (Sound) ข้อความ (Text) ภาพกราฟิกต่างๆ (Graphics) รวมถึงสื่อชนิดอื่นๆ ด้วย ดังภาพที่ 6.9



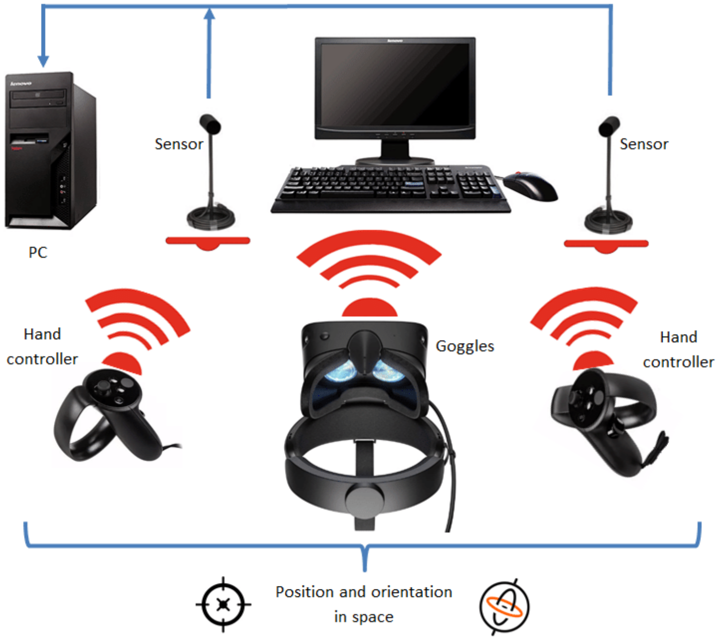
ภาพที่ 6.9 มัลติมีเดียและไฮเปอร์เท็กซ์

3. ระบบความเป็นจริงเสมือน (Virtual Reality System)

ระบบความเป็นจริงเสมือน (Virtual Reality System) คือหนึ่งในรูปแบบการสื่อสารกับผู้ใช้ที่ได้รับความนิยมอย่างมากในปัจจุบัน การแสดงผลในมิติสามมิติเป็นส่วนหนึ่งของระบบนี้ที่มีความสำคัญ เพื่อสร้างสภาพแวดล้อมที่ถูกจำลองขึ้นโดยคอมพิวเตอร์ (Computer-Simulated Environment) ผู้ใช้ระบบสามารถสัมผัสและเข้าไปอยู่ในสภาพแวดล้อมที่ถูกสร้างขึ้นได้ โดยอุปกรณ์พิเศษที่ถูกพัฒนาขึ้นมาเป็นส่วนสำคัญในกระบวนการนี้

อุปกรณ์พิเศษเหล่านี้รวมถึงถุงมือ แว่นตา หรือหมวกที่ถูกออกแบบมาเพื่อให้ผู้ใช้สามารถสื่อสารและปฏิสัมพันธ์กับสภาพแวดล้อมที่จำลองได้ ระบบจะบันทึกการเคลื่อนไหว การสื่อสารเสียง และการรับรู้ความรู้สึกของผู้ใช้ เช่น เมื่อผู้ใช้เคลื่อนไปทางขวา ระบบจะสร้างภาพที่เคลื่อนไหวไปทางขวาตามการเคลื่อนไหวของผู้ใช้ นั่นหมายความว่าผู้ใช้จะมีความรู้สึกคล้ายกับอยู่ในโลกความเป็นจริง

ตัวอย่างของอุปกรณ์สำหรับระบบความเป็นจริงเสมือนที่ทันสมัยได้แก่ แว่นตาเสมือนจริง (Virtual Reality Headset) ที่สร้างประสบการณ์สามมิติที่มีความคล้ายคลึงกับโลกความเป็นจริง หรือถุงมือที่สามารถตรวจจับการเคลื่อนไหวของมือและนิ้ว เพื่อให้ผู้ใช้สามารถปฏิสัมพันธ์กับสภาพแวดล้อมที่ถูกสร้างขึ้นได้อย่างเหมือนจริง การพัฒนาระบบความเป็นจริงเสมือนที่ใช้เทคโนโลยีทันสมัยเช่นนี้มีบทบาทสำคัญในการสร้างประสบการณ์ที่น่าทึ่งและนำผู้ใช้เข้าสู่โลกความเป็นจริงอย่างมีความสมจริง ดังภาพที่ 6.10



ภาพที่ 6.10 ระบบความเป็นจริงเสมือน
(ที่มา : Burghardt et al., 2020)

ในโลกที่เติบโตและพัฒนาการเทคโนโลยีอย่างรวดเร็ว เทคโนโลยีเสมือนจริงก็ได้รับความสนใจและพัฒนาขึ้นมาอย่างต่อเนื่อง เทคโนโลยีเสมือนจริงมีหลายรูปแบบที่มีความแตกต่างกันอย่างเป็นนัยสำคัญ ซึ่งรวมถึง Augmented Reality (AR) Virtual Reality (VR) และ Mixed Reality (MR) ซึ่งเป็นเทคโนโลยีที่มีความสำคัญในการเปลี่ยนแปลงวิธีการสื่อสารและประสบการณ์ของผู้ใช้ในโลกดิจิทัลได้อย่างมีนัยสำคัญ ได้แก่

1) Augmented Reality (AR) เป็นเทคโนโลยีที่นำสิ่งต่าง ๆ ในโลกจริงมาผสมผสานกับสิ่งเสมือนจริงผ่านทางอุปกรณ์อิเล็กทรอนิกส์ เช่น สมาร์ทโฟนหรือแว่นตา AR ผู้ใช้จะเห็นภาพข้อมูลเสริมเข้ามาในสภาพแวดล้อมจริง เช่น การใช้แอปพลิเคชัน AR เพื่อแสดงข้อมูลเพิ่มเติมเมื่อเห็นรูปภาพในโลกจริง หรือการใช้ AR ในการแสดงนาฬิกาหรือข้อมูลนำทางขณะที่เดินทาง

2) Virtual Reality (VR) เป็นเทคโนโลยีที่สร้างสภาพแวดล้อมเสมือนจริงที่ผู้ใช้สามารถเข้าไปในโลกเสมือนจริงหรือประสบการณ์เหมือนจริงผ่านทางอุปกรณ์ VR เช่น หูฟัง VR และแว่น VR ผู้ใช้จะได้รับประสบการณ์ที่คล้ายกับการอยู่ในโลกเสมือนจริง และเป็นส่วนหนึ่งในแบบจำลองสภาพแวดล้อมหรือสร้างเกมหรือประสบการณ์ต่าง ๆ ได้อย่างสมจริง

3) Mixed Reality (MR) เป็นการผสมผสานระหว่างโลกเสมือนจริงและโลกจริงในลักษณะที่สร้างขึ้นผ่านทางอุปกรณ์พิเศษ เช่น แว่น MR ผู้ใช้จะได้เห็นวัตถุและสิ่งต่าง ๆ ที่ถูกจำลองขึ้นมาในสภาพแวดล้อมจริง และ

สามารถปฏิสัมพันธ์กับวัตถุดังกล่าวได้ ตัวอย่างการใช้ MR คือการใช้งานในสภาพแวดล้อมที่มีวัตถุที่เป็นจริงในขณะที่มีวัตถุเสมือนจริงผสมเข้าไป

สรุปเทคโนโลยีเสมือนจริงมีหลายรูปแบบที่ให้ประสบการณ์ที่แปลกใหม่และสร้างโอกาสใหม่ในการสื่อสารและประสบการณ์ในโลกดิจิทัลไม่ว่าจะเป็น Augmented Reality (AR) ที่เพิ่มข้อมูลในโลกจริง หรือ Virtual Reality (VR) ที่ให้ประสบการณ์ในโลกเสมือนจริง หรือ Mixed Reality (MR) ที่ผสมผสานทั้งสองโลกเข้าด้วยกัน เทคโนโลยีเหล่านี้มีศักยภาพในการเปลี่ยนแปลงวิธีการสื่อสารและประสบการณ์ของเราในอนาคต

สรุปท้ายบทที่ 6

ระบบสารสนเทศจำเป็นต้องถูกออกแบบเพื่อให้มีการโต้ตอบที่มีประสิทธิภาพกับผู้ใช้ เนื่องจากผู้ใช้เป็นปัจจัยสำคัญในกระบวนการปฏิบัติงานและการใช้งานระบบ การออกแบบการโต้ตอบระหว่างผู้ใช้และระบบมีความสำคัญเพื่อให้การใช้งานเป็นไปอย่างมีประสิทธิภาพและเป็นมิตรกับผู้ใช้ ในการเลือกใช้อุปกรณ์สื่อสารกับผู้ใช้ควรเลือกใช้อุปกรณ์ที่เหมาะสมเพื่อการโต้ตอบ เพื่อเสริมสร้างความพึงพอใจให้กับผู้ใช้งาน ยูสเซอร์อินเตอร์เฟซมีหลายแบบที่เหมาะสมกับบริบทและลักษณะงาน ได้แก่ การอินเตอร์เฟซด้วยภาษาธรรมชาติ การอินเตอร์เฟซด้วยคำถามและคำตอบ การอินเตอร์เฟซด้วยเมนู การอินเตอร์เฟซด้วยชุดคำสั่ง และการอินเตอร์เฟซแบบกราฟิก (GUI) การเลือกรูปแบบยูสเซอร์อินเตอร์เฟซที่เหมาะสมกับรูปแบบการใช้งานและลักษณะงานจะสร้างประสิทธิภาพและความสะดวกในการทำงานของผู้ใช้

การออกแบบการโต้ตอบด้วยเทคนิคสตอรีบอร์ดเป็นวิธีที่ช่วยให้เห็นขั้นตอนการโต้ตอบอย่างละเอียดโดยการนำเสนอเหตุการณ์และลำดับการทำงานอย่างสรุปในรูปแบบย่อหน้า เทคนิคนี้ช่วยให้ผู้ออกแบบและผู้ใช้งานเข้าใจโครงสร้างของการโต้ตอบในแต่ละเหตุการณ์ได้อย่างชัดเจน ผลที่ได้คือ การสร้างโปรโตไทป์ที่สามารถช่วยในการวางโครงสร้างระบบได้รวดเร็ว โดยแผนผังโครงสร้างจะเน้นความสัมพันธ์ระหว่างโมดูลและข้อมูลที่ส่งผ่านระหว่างกัน เพื่อให้ได้โครงสร้างที่มีความสัมพันธ์แน่นและสามารถปรับปรุงได้ง่าย หลักการออกแบบที่ดีคือให้โมดูลมีความสัมพันธ์แน่นและเป็นหนึ่งเดียว (High Cohesion) และมีความสัมพันธ์ระหว่างโมดูลที่แยกกันเป็นระดับหลวมๆ (Loosely Coupled)

คำถามท้ายบทที่ 6

1. อธิบายวัตถุประสงค์และความสำคัญของการออกแบบระบบ
2. ยกตัวอย่างสื่อและเทคโนโลยีที่ใช้ในการแสดงผลลัพธ์และการนำเข้าข้อมูล ด้านละ 3 ตัวอย่าง
3. องค์ประกอบของส่วนติดต่อผู้ใช้มีกี่อย่าง จงอธิบาย
4. ประเภทของส่วนติดต่อผู้ใช้มีกี่อย่าง จงอธิบาย
5. ยกตัวอย่างหลักการออกแบบของจาคอป (Jakob's Principle of Design) มา 5 ข้อ
6. ยกตัวอย่างเทคโนโลยีที่ใช้การโต้ตอบด้วยภาษามนุษย์ (Natural Language Interaction) ในปัจจุบันมา 5 ตัวอย่าง

บทที่ 7
ระบบขายหน้าร้าน (Point of Sale System)

เนื้อหาในบทนี้จะเป็นการประยุกต์ใช้งานจริงโดยจะยกตัวอย่างระบบขายหน้าร้าน (Point of Sale System) ซึ่งเป็นระบบที่ใช้ในธุรกิจขายปลีกที่ได้รับความนิยมมากในปัจจุบัน ระบบ POS เป็นเครื่องมือสำคัญที่ช่วยให้ธุรกิจขายปลีกทำงานได้อย่างมีประสิทธิภาพและมีการควบคุมที่ดีต่อการจัดการธุรกิจและสร้างรายงานที่จำเป็นสำหรับการตัดสินใจและวางแผนในอนาคตได้ โดยทั่วไประบบ POS จะมีความสามารถในการจัดการข้อมูลการขาย การชำระเงิน สต็อกสินค้า และช่วยในการบันทึกรายงานการขายและการรายงานอื่น ๆ ที่เกี่ยวข้องกับธุรกิจ คุณสมบัติสำคัญของระบบ POS ที่ต้องมี เช่น การทำธุรกรรมการขาย จัดการสต็อก การบันทึกการชำระเงิน ออกใบเสร็จรับเงิน รายงานการสั่งซื้อ การจัดการสมาชิก ควบคุมการเข้าถึง เป็นต้น

ซึ่งในบทนี้จะยกตัวอย่างเริ่มต้นจากการรวบรวมความต้องการใช้งานของผู้ใช้ เพื่อนำมาใช้เป็นข้อมูลพื้นฐานในการวิเคราะห์ความต้องการของระบบ จากนั้นจึงเป็นการวิเคราะห์และออกแบบระบบตามวงจรการพัฒนาระบบ ประกอบด้วยหัวข้อ ดังนี้

1. การรวบรวมความต้องการใช้งานของผู้ใช้งาน
2. วิเคราะห์ความต้องการของระบบ
3. การสร้างแผนภาพกระแสข้อมูล
4. คำอธิบายกระบวนการแผนภาพกระแสข้อมูล
5. การสร้างแผนภาพ E-R
6. โครงสร้างฐานข้อมูล

การรวบรวมความต้องการใช้งานของผู้ใช้งาน

การรวบรวมความต้องการใช้งานของผู้ใช้งาน จะเริ่มต้นจากการศึกษากระบวนการทำงานและกฎเกณฑ์ของธุรกิจของงานนั้น ๆ ในกรณีศึกษานี้จะพัฒนาระบบขายหน้าร้านให้กับร้านขายสินค้าเบ็ดเตล็ด ซึ่งจะมีเครื่องคิดเงินติดกับเครื่องสแกนบาร์โค้ดและติดตั้งระบบขายหน้าร้าน POS ไว้ในเครื่อง ดังนั้นจะต้องศึกษากระบวนการทำงานของการขายหน้าร้านจากระบบเดิม เพื่อวิเคราะห์ความต้องการของระบบที่จะพัฒนาใหม่ กระบวนการทำงานและกฎเกณฑ์ทางธุรกิจของการขายหน้าร้าน จะประกอบด้วยขั้นตอน ดังนี้

1. ลูกค้าเข้าร้านขายสินค้าเบ็ดเตล็ดเลือกสินค้าต่าง ๆ ภายในร้าน เมื่อเลือกสินค้าที่ต้องการครบแล้ว จะนำสินค้ามาที่จุดขาย หรือ จุดชำระสินนค้าของร้าน ซึ่งจะมีพนักงานขายสินค้าประจำอยู่ที่จุดขาย

2. พนักงานที่ประจำอยู่ที่จุดขาย หรือจุดชำระสินค้ากับเครื่องคิดเงิน จะต้องมีการยืนยันตัวตนในการขายสินค้าแต่ละครั้งให้ลูกค้า

3. พนักงานขายจะตรวจสอบสินค้าที่ลูกค้านำมาที่จุดขาย จากนั้นก็จะนับจำนวนสินค้าและดูราคาสินค้าแต่ละชิ้นพร้อมกันไปด้วย พร้อมกับกดเครื่องคิดเลขเพื่อบวกราคาสินค้าแต่ละชิ้นจนได้ยอดรวมชำระสินค้า จากนั้นพนักงานก็จะแจ้งราคารวมที่ต้องชำระให้ลูกค้าทราบ ลูกค้าก็จะจ่ายเงินสดให้พนักงาน ถ้าจำนวนเงินมากกกว่าราคาที่ต้องชำระ พนักงานก็จะทอนเงินสดให้ลูกค้า พร้อมเขียนยอดรวมราคาสินค้าในใบเสร็จให้ลูกค้าด้วย

4. เจ้าของร้านหรือพนักงานขายจะนับสินค้าที่เพิ่มเข้ามาใหม่ในร้าน และตรวจสอบจำนวนสินค้าคงเหลือในร้านทุกวัน

5. ในแต่ละวันเจ้าของร้านจะมาตรวจสอบยอดขายที่ขายสินค้าไป และตรวจสอบรายรับแต่ละวัน โดยดูจากสำเนาใบเสร็จที่เขียนให้กับลูกค้ากับจำนวนเงินสดที่มีแต่ละวัน

การวิเคราะห์ความต้องการของระบบ

ขั้นตอนถัดมาจะเป็นการวิเคราะห์ความต้องการของระบบจากกระบวนการทำงานและกฎเกณฑ์ทางธุรกิจของการขายหน้าร้านที่ได้กล่าวไว้แล้วข้างต้น ซึ่งจะสรุปเป็นประเด็นหลัก ดังนี้

1. ระบบขายหน้าร้านควรพัฒนาให้ทำงานบนเครื่องคิดเงินได้และนำข้อมูลสินค้า หรือรายการสั่งซื้อสินค้าเข้าสู่ระบบด้วยการสแกนบาร์โค้ดได้ ดังนั้นสามารถพัฒนาระบบให้ทำงานแบบออฟไลน์ หรือการใช้งานเครื่องเดียว (Standalone) ได้

2. ระบบจะต้องมีฟังก์ชันการยืนยันตัวตนเข้าสู่ระบบของพนักงานหรือเจ้าของร้าน เพื่อเก็บประวัติการขายสินค้า หรือประวัติการใช้งานระบบได้ เพื่อเจ้าของร้านจะสามารถตรวจสอบข้อมูลได้ภายหลัง

3. ระบบจะต้องมีฟังก์ชันรับข้อมูลจากเครื่องสแกนบาร์โค้ดได้ เมื่อลูกค้านำสินค้ามาที่จุดชำระสินค้าพนักงานขายจะใช้เครื่องสแกนบาร์โค้ดเพื่อสแกนรหัสสินค้าเพื่อนำสินค้าเข้ารายการสั่งซื้อ

4. ระบบจะต้องสามารถที่จะสั่งซื้อสินค้าได้มากกว่า 1 ชิ้นได้ และสามารถคำนวณราคาสินค้าทั้งหมดเป็นจำนวนเงินที่ต้องชำระในแต่ละครั้งได้อย่างถูกต้อง

5. ระบบต้องมีฟังก์ชันในการรับเงินสดจากลูกค้า และคำนวณเงินทอนให้ลูกค้าได้อย่างถูกต้อง เพื่อให้พนักงานนำเงินสดในเครื่องคิดเงินทอนเงินให้ลูกค้าได้อย่างถูกต้อง

6. ระบบจะต้องมีฟังก์ชันในการพิมพ์ใบเสร็จรับเงินออกทางเครื่องพิมพ์ให้ลูกค้าในการสั่งซื้อสินค้าแต่ละครั้งได้อย่างถูกต้อง

7. ระบบจะต้องมีฟังก์ชันในการเพิ่มสินค้า เช่น ชื่อสินค้า รายละเอียดสินค้า ราคาสินค้า และเพิ่มจำนวนสินค้า เป็นต้น เข้าสู่ระบบได้ เมื่อพนักงานขายหรือเจ้าของร้านสั่งซื้อสินค้าเข้ามาเพิ่มในร้าน ระบบต้องสามารถเพิ่มสินค้าใหม่เข้าระบบได้ และเพิ่มจำนวนสินค้าคงเหลือได้

8. ระบบต้องมีฟังก์ชันสามาถตรวจสอบประวัติการสั่งซื้อสินค้าของลูกค้าได้

9. ระบบต้องมีฟังก์ชันสามารถตรวจสอบสินค้าคงเหลือภายในร้านได้

10. ระบบต้องมีฟังก์ชันแสดงรายงานยอดสั่งซื้อสินค้าเป็นรายวัน เดือน ปี ได้ และสามารถพิมพ์ยอดขายได้

จะเห็นได้ว่าฟังก์ชันแสดงรายงานยอดสั่งซื้อสินค้าเป็นรายวัน เดือน ปีได้ เกิดขึ้นจากขั้นตอนการวิเคราะห์ความต้องการของระบบ โดยเพิ่มเติมจากกระบวนการทำงานและกฎเกณฑ์ทางธุรกิจของการขายหน้าร้านที่ได้จากการรวบรวมข้อมูลพื้นฐาน เนื่องจากระบบเดิมไม่ใช้ได้คอมพิวเตอร์ เมื่อลูกค้ามาสั่งซื้อสินค้าแล้วเจ้าของร้านต้องการตรวจสอบยอดขายสินค้าจะสามารถตรวจสอบได้เป็นรายวันเท่านั้นเนื่องจากใช้การจดบันทึกทั้งหมด

ดังนั้น เมื่อนักวิเคราะห์ระบบได้วิเคราะห์ระบบงานในภาพรวมแล้ว จึงพิจารณาได้ว่าระบบสารสนเทศที่จะพัฒนาขึ้นสามารถที่จะอำนวยความสะดวกแก่ผู้ใช้เพิ่มเติมในประเด็นใดได้บ้าง

การสร้างแผนภาพกระแสข้อมูล

การสร้างแผนภาพกระแสข้อมูล (Data Flow Diagram) ได้อธิบายไว้แล้วอย่างละเอียดในบทที่ 4 ซึ่งสามารถดำเนินตามขั้นตอนการสร้างแผนภาพกระแสข้อมูล ดังนี้

1. การวิเคราะห์องค์ประกอบของแผนภาพกระแสข้อมูล

1.1 ค้นหาคำสำคัญ จากกระบวนการและกฎเกณฑ์ทางธุรกิจของระบบขายหน้าร้านแต่ละข้อและผลของการวิเคราะห์ความต้องการของระบบในหัวข้อที่ผ่านมา ดังนั้นต้องค้นหาคำสำคัญเพื่อจัดกลุ่มองค์ประกอบของแผนภาพกระแสข้อมูล ดังนี้

1) ค้นหาคำสำคัญที่น่าจะเป็น เอ็นทิตี้ภายนอก

2) ค้นหาคำสำคัญที่น่าจะเป็น กระแสข้อมูลนำเข้า

3) ค้นหาคำสำคัญที่น่าจะเป็น กระแสข้อมูลสำหรับการประมวลผล

4) ค้นหาคำสำคัญที่น่าจะเป็น กระแสข้อมูลผลลัพธ์

5) ค้นหาคำสำคัญที่น่าจะปีน แฟ้มข้อมูล

1.2 การจัดกลุ่มคำสำคัญ เพื่อใช้เป็นองค์ประกอบในการสร้างแผนภาพกระแสข้อมูล ดังตารางที่ 7.1 ซึ่งในกรณีที่มีคำสำคัญหลายคำที่สื่อความหมายอย่างเดียวกัน นักวิเคราะห์ระบบควรตัดสินใจเลือกใช้คำใดคำหนึ่งที่สามารถสื่อสารให้ผู้ใช้เข้าใจถึงระบบที่กำลังจะพัฒนาได้อย่างชัดเจนที่สุด

ตารางที่ 7.1 คำสำคัญที่น่าจะเป็นองค์ประกอบของแผนภาพกระแสข้อมูล

องค์ประกอบของแผนภาพกระแสข้อมูล	คำสำคัญ
เอ็นทิตี้ภายนอก	- พนักงานขาย - เจ้าของร้าน
กระแสข้อมูลนำเข้า	- ข้อมูลยืนยันตัวตน - ข้อมูลการสั่งซื้อ - ข้อมูลสินค้า - ข้อมูลจำนวนเงิน - ข้อมูลเพิ่มสินค้า
กระแสข้อมูลสำหรับการประมวลผล	- ข้อมูลพนักงาน เจ้าของร้าน - ข้อมูลเข้าสู่ระบบ - ข้อมูลสั่งซื้อสินค้า - ข้อมูลเพิ่มสินค้า - ข้อมูลตรวจสอบสินค้าคงเหลือ - ข้อมูลรายงานยอดสั่งซื้อ
กระแสข้อมูลผลลัพธ์	- ข้อมูลผลการเข้าสู่ระบบ - ข้อมูลผลการสั่งซื้อสินค้า - ข้อมูลผลการเพิ่มสินค้า - ข้อมูลผลการตรวจสอบสินค้าคงเหลือ - ข้อมูลผลการตรวจสอบรายงานยอดสั่งซื้อ
แฟ้มข้อมูล	- แฟ้มข้อมูลพนักงาน - แฟ้มข้อมูลเจ้าของร้าน - แฟ้มข้อมูลสินค้า - แฟ้มข้อมูลประเภทสินค้า - แฟ้มข้อมูลการสั่งซื้อ - แฟ้มข้อมูลรายละเอียดการสั่งซื้อ

ดังนั้นจึงสามารถแสดงความสัมพันธ์ขององค์ประกอบต่าง ๆ ของแผนภาพกระแสข้อมูลสำหรับกระบวนการทำงานระบบ POS ที่ผ่านการวิเคราะห์องค์ประกอบของแผนภาพกระแสข้อมูลได้ดังตารางที่ 7.2

ตารางที่ 7.2 ความสัมพันธ์ขององค์ประกอบต่างๆ สำหรับแต่ละกระบวนการทำงาน

กระบวนการทำงาน	**กระแสข้อมูลที่สัมพันธ์กับองค์ประกอบ**		
	เอ็นทิตี้ภายนอก	**แหล่งจัดเก็บข้อมูล**	**กระบวนการทำงานอื่นๆ**
1. จัดการสมาชิก	- เจ้าของร้าน - พนักงานขาย	- แฟ้มข้อมูลพนักงาน - แฟ้มข้อมูลเจ้าของร้าน	เพิ่มข้อมูลสมาชิก
2. เข้าสู่ระบบ	- เจ้าของร้าน - พนักงานขาย	- แฟ้มข้อมูลพนักงาน - แฟ้มข้อมูลเจ้าของร้าน	ตรวจสอบสิทธิ์การใช้งานด้วยรหัสเข้าใช้งาน และรหัสผ่าน
3. ค้นหาข้อมูลสินค้า	- พนักงานขาย	- แฟ้มข้อมูลสินค้า	ค้นหาข้อมูลสินค้าจากรหัสสินค้า
4. สั่งซื้อสินค้า	- พนักงานขาย	- แฟ้มข้อมูลสินค้า - แฟ้มข้อมูลประเภทสินค้า - แฟ้มข้อมูลการสั่งซื้อ	- คำนวณยอดสั่งซื้อที่ลูกค้าต้องชำระ - รับเงินสอบคำนวณเงินทอนให้ลูกค้า
5. จัดการสินค้า	- เจ้าของร้าน	- แฟ้มข้อมูลสินค้า - แฟ้มข้อมูลประเภทสินค้า	- เพิ่มสินค้าใหม่ - แก้ไขสินค้า - เพิ่มจำนวนสินค้าคงเหลือ
6. ตรวจสอบการสั่งซื้อสินค้า	- เจ้าของร้าน	- แฟ้มข้อมูลการสั่งซื้อ - แฟ้มข้อมูลรายละเอียดการสั่งซื้อ	- แสดงประวัติการสั่งซื้อของลูกค้า - แสดงประวัติการขายของพนักงานขายแต่ละการสั่งซื้อ

ตารางที่ 7.2 ความสัมพันธ์ขององค์ประกอบต่างๆ สำหรับแต่ละกระบวนการทำงาน (ต่อ)

กระบวนการทำงาน	**กระแสข้อมูลที่สัมพันธ์กับองค์ประกอบ**		
	เอนทิตีภายนอก	**แหล่งจัดเก็บข้อมูล**	**กระบวนการทำงานอื่นๆ**
7. ตรวจสอบสินค้าคงเหลือ	- เจ้าของร้าน	- แฟ้มข้อมูลสินค้า - แฟ้มข้อมูลประเภทสินค้า - แฟ้มข้อมูลการสั่งซื้อ - แฟ้มข้อมูลรายละเอียดการสั่งซื้อ	- แสดงจำนวนสินค้าของสินค้าแต่ละรายการ
8. รายงานยอดสั่งซื้อ	- เจ้าของร้าน	- แฟ้มข้อมูลการสั่งซื้อ - แฟ้มข้อมูลรายละเอียดการสั่งซื้อ	- แสดงรายงานยอดขายรายวัน รายเดือน รายปี - พิมพ์รายงานออกมาได้

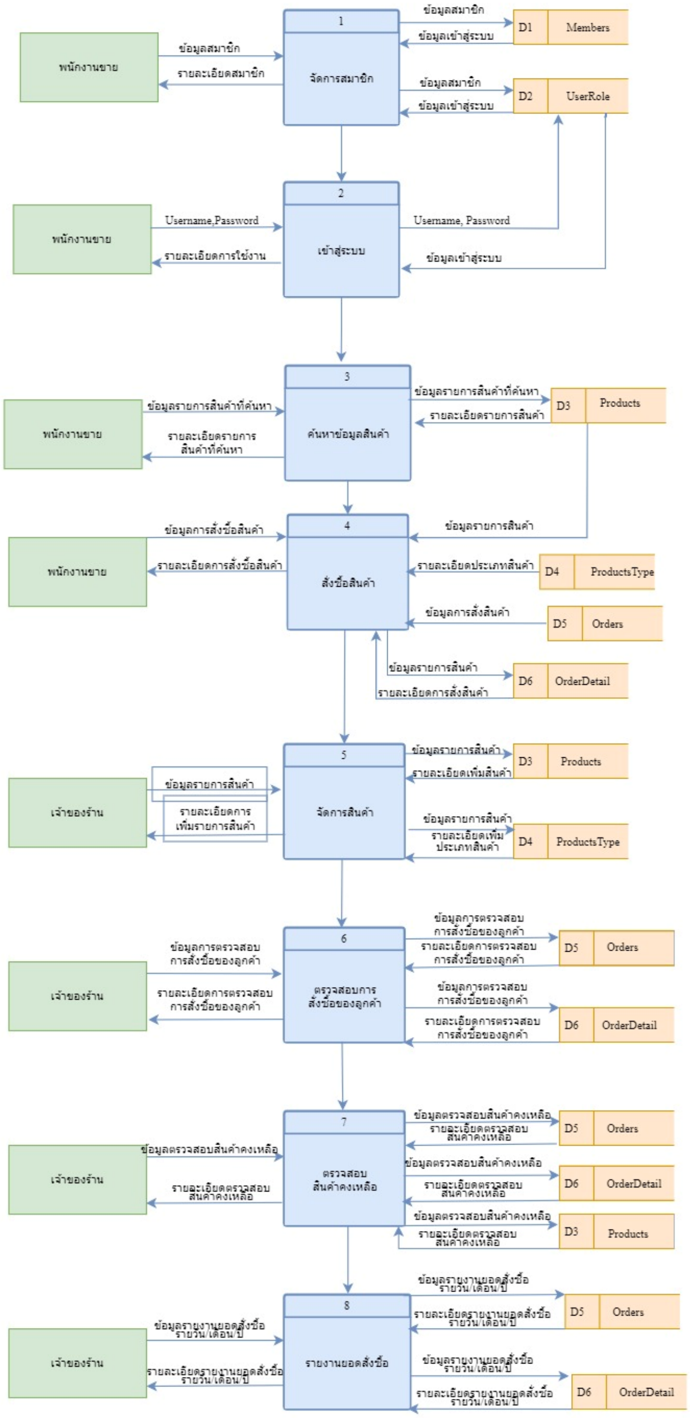
2. การสร้างแผนภาพบริบท

จากตารางที่ 7.2 จะนำเอาองค์ประกอบที่เป็นเอ็นทิตี้ ภายนอกและกระแสข้อมูลที่เกี่ยวข้องกับเอ็นทิตี้ภายนอกมาใช้สร้างแผนภาพบริบท แสดงดังภาพที่ 7.1

ภาพที่ 7.1 แผนภาพบริบทของระบบ POS

3. การสร้างแผนภาพระดับที่ 0

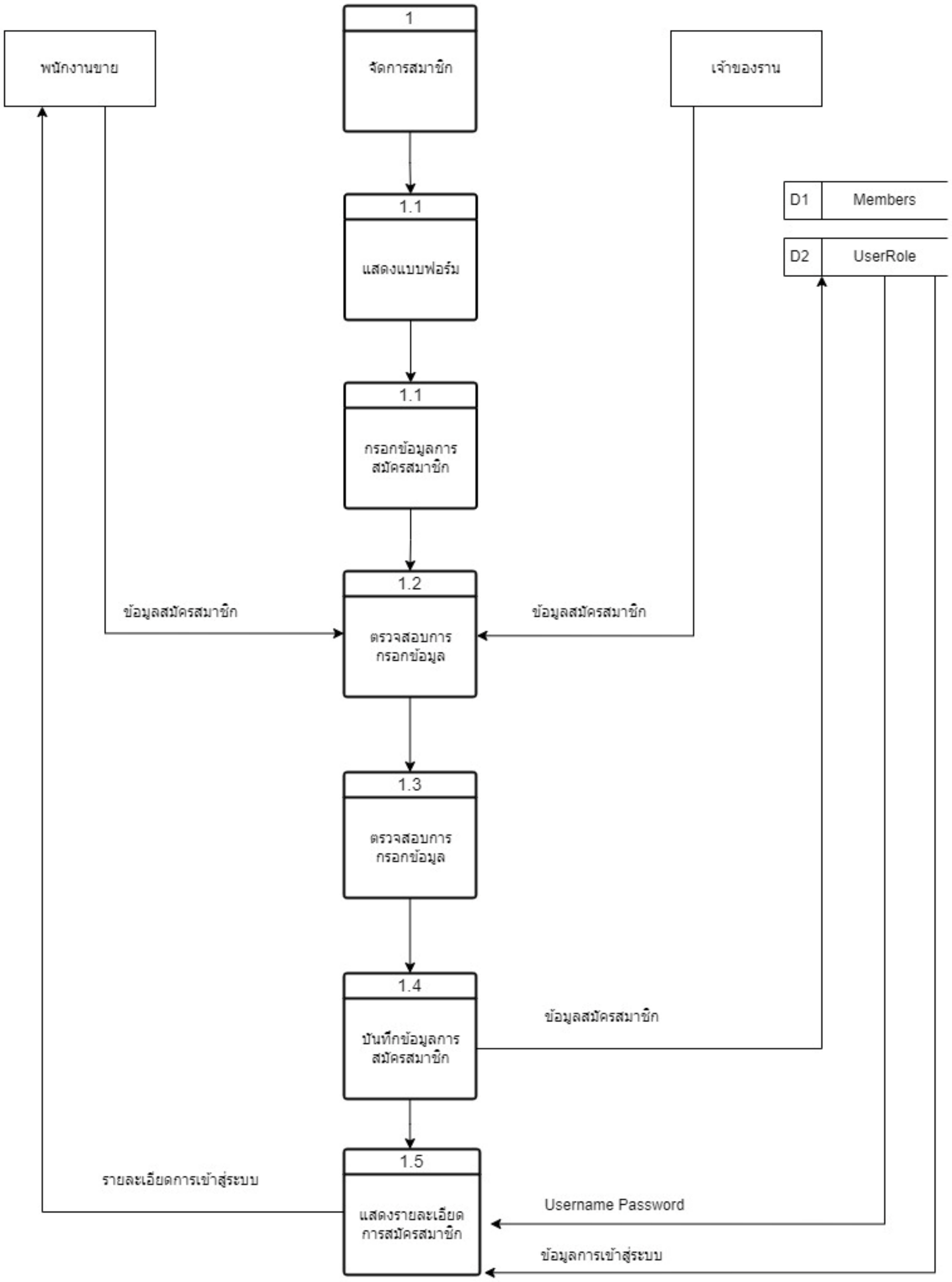
แผนภาพระดับที่ 0 ของระบบ POS ดังแสดงในภาพที่ 7.2



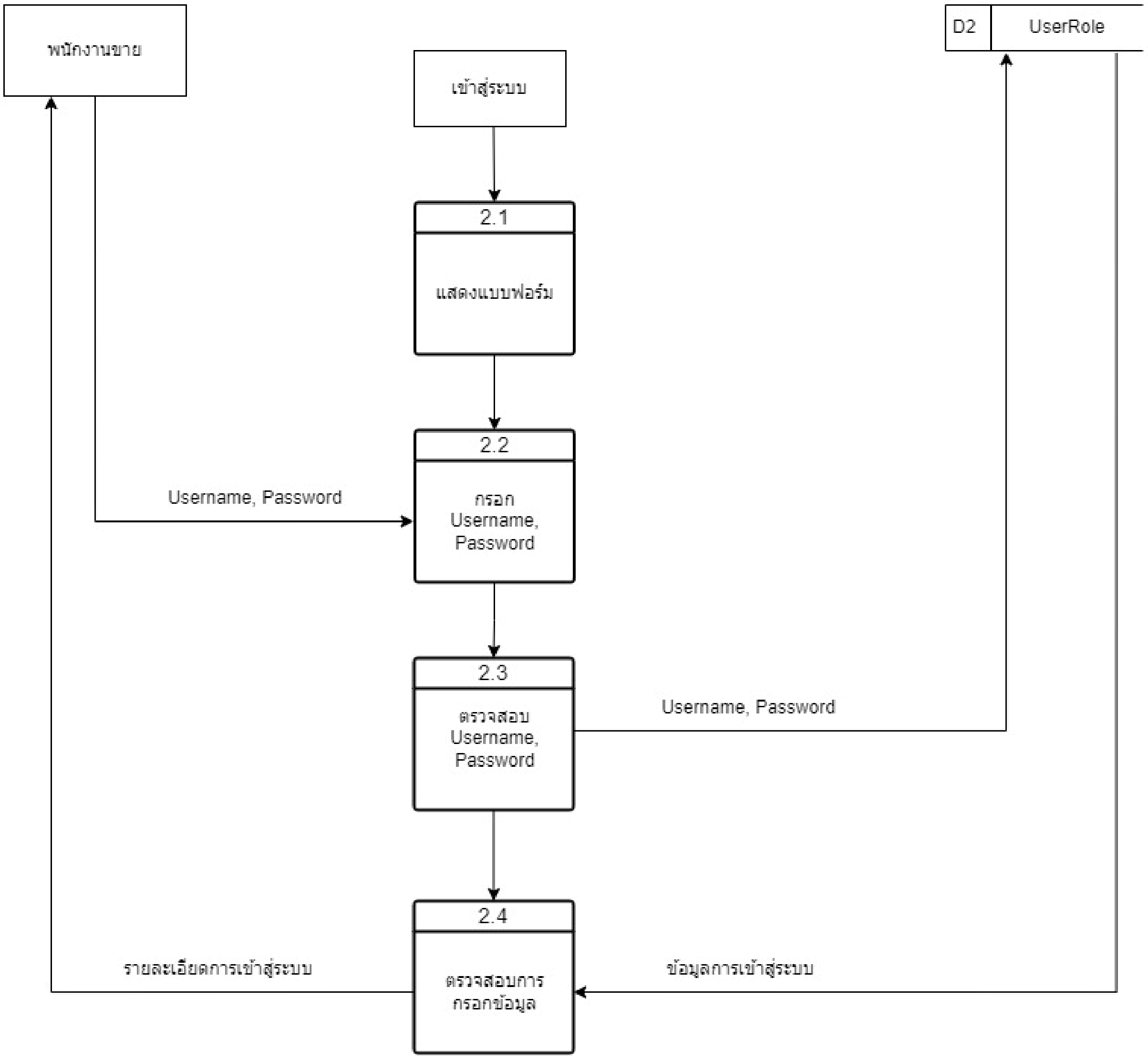
ภาพที่ 7.2 แผนภาพระดับที่ 0 ของระบบ POS

4. การสร้างแผนภาพระดับที่ 1

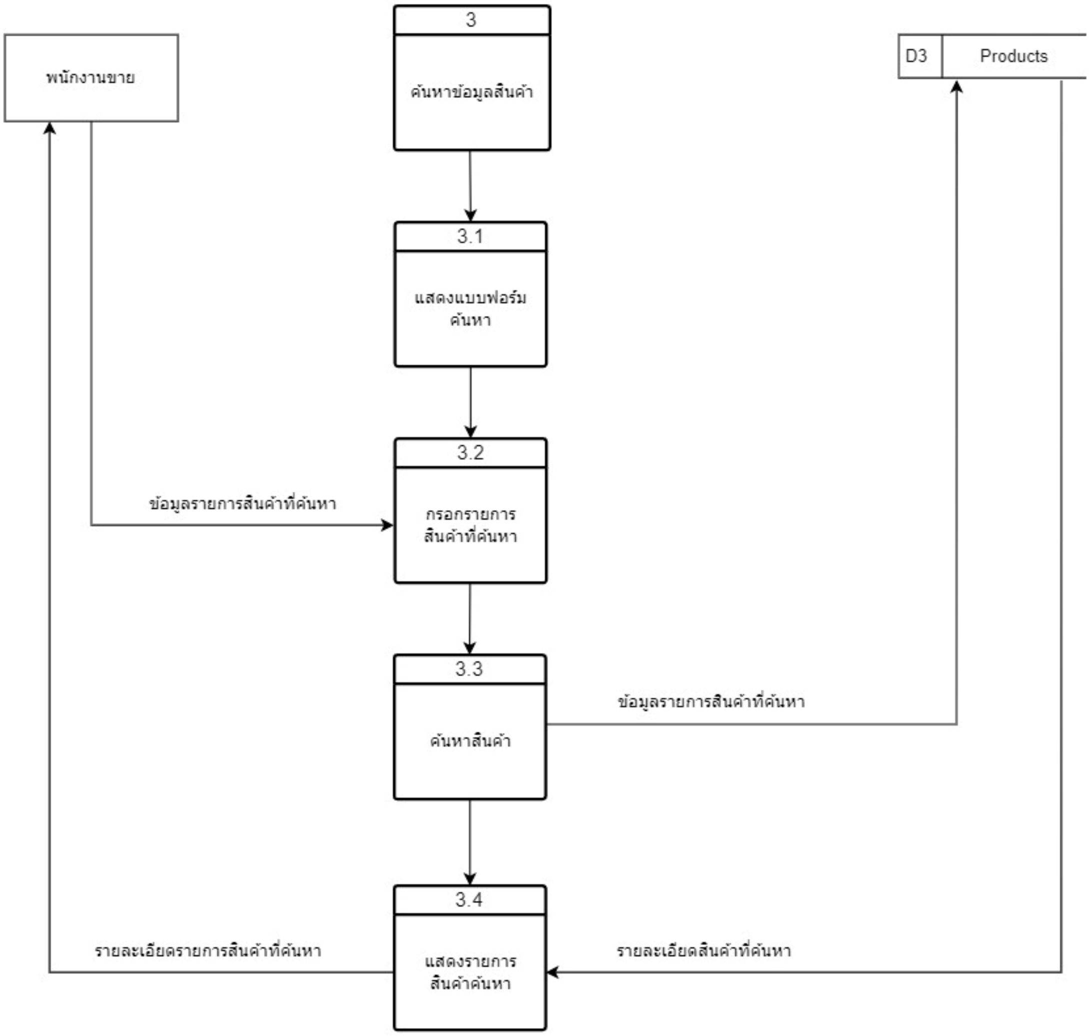
แผนภาพระดับที่ 1 ของกระบวนการที่ 1 จัดการสมาชิก ของระบบ POS ดังแสดงในภาพที่ 7.3



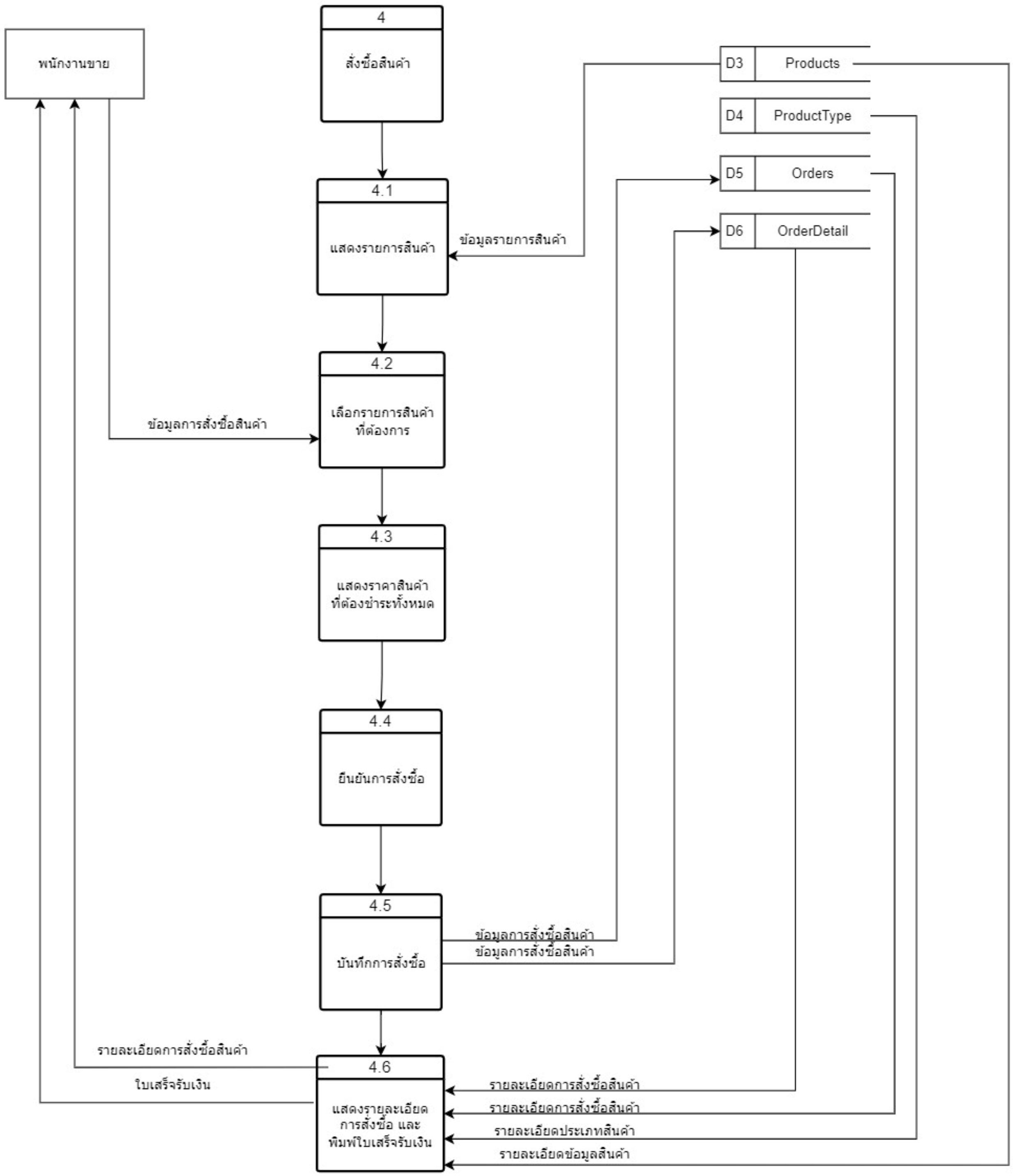
ภาพที่ 7.3 แผนภาพระดับที่ 1 ของกระบวนการที่ 1 จัดการสมาชิก



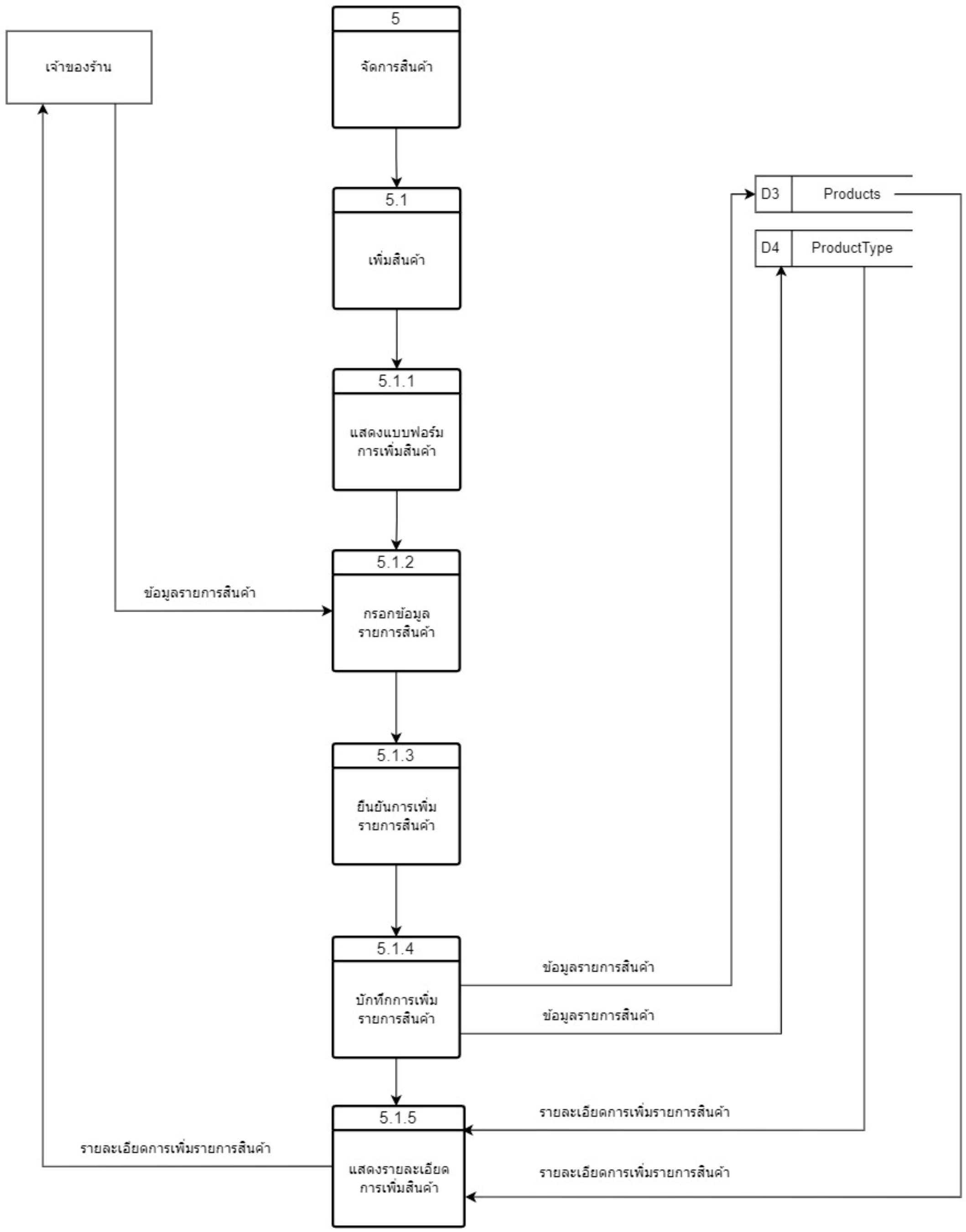
ภาพที่ 7.4 แผนภาพระดับที่ 1 ของกระบวนการที่ 2 เข้าสู่ระบบ



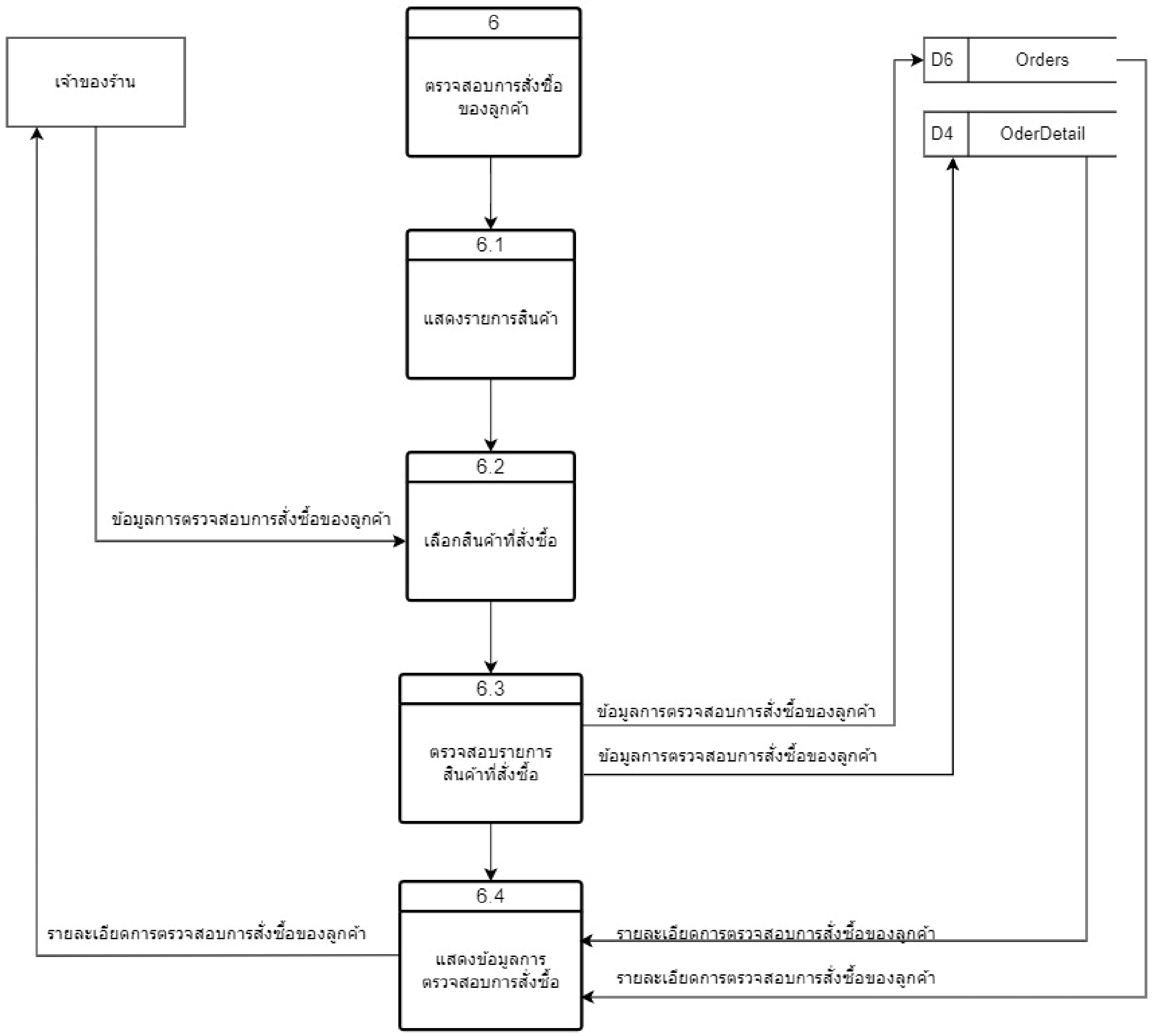
ภาพที่ 7.5 แผนภาพระดับที่ 1 ของกระบวนการที่ 3 ค้นหาข้อมูลสินค้า



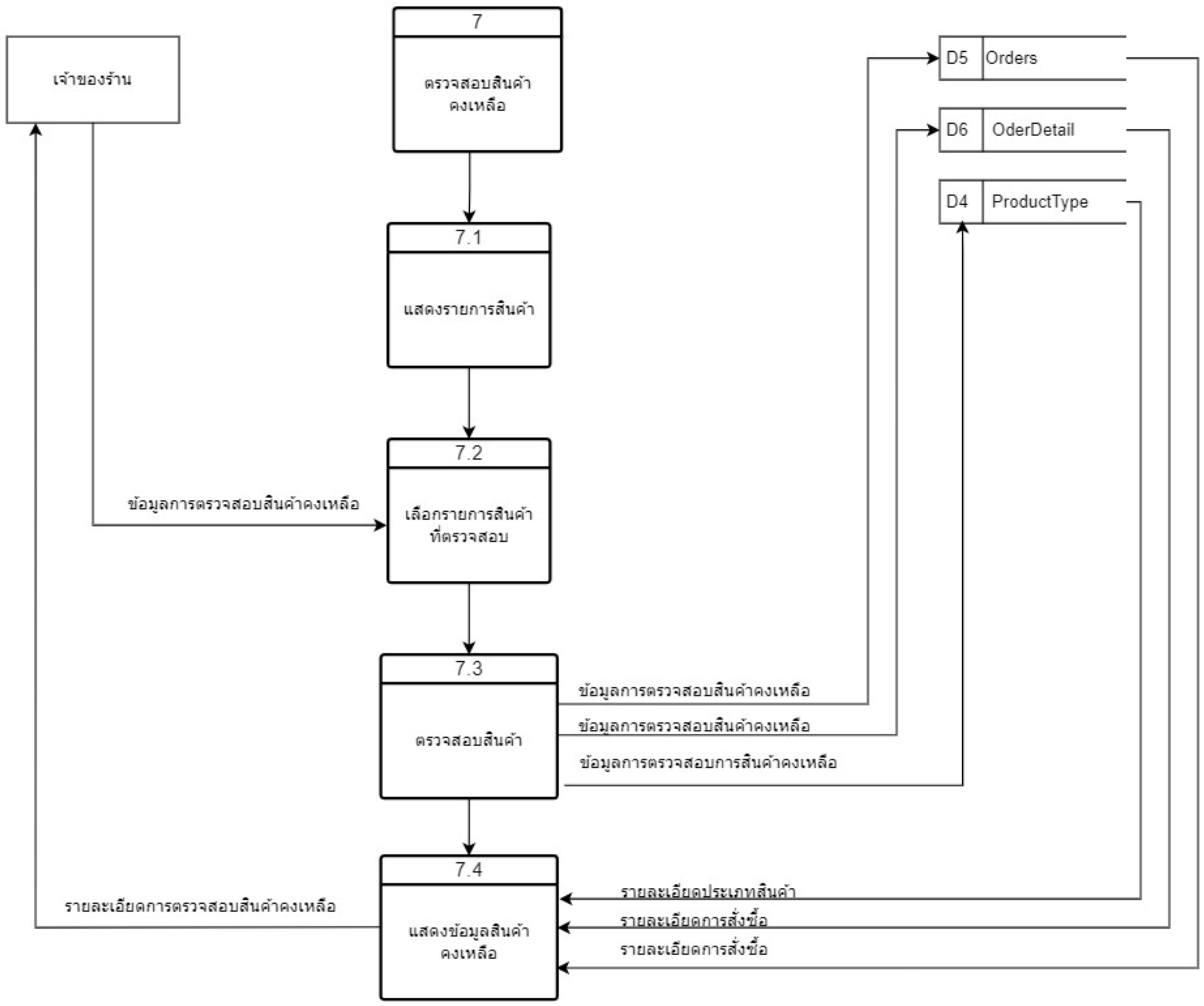
ภาพที่ 7.6 แผนภาพระดับที่ 1 ของกระบวนการที่ 4 สั่งซื้อสินค้า



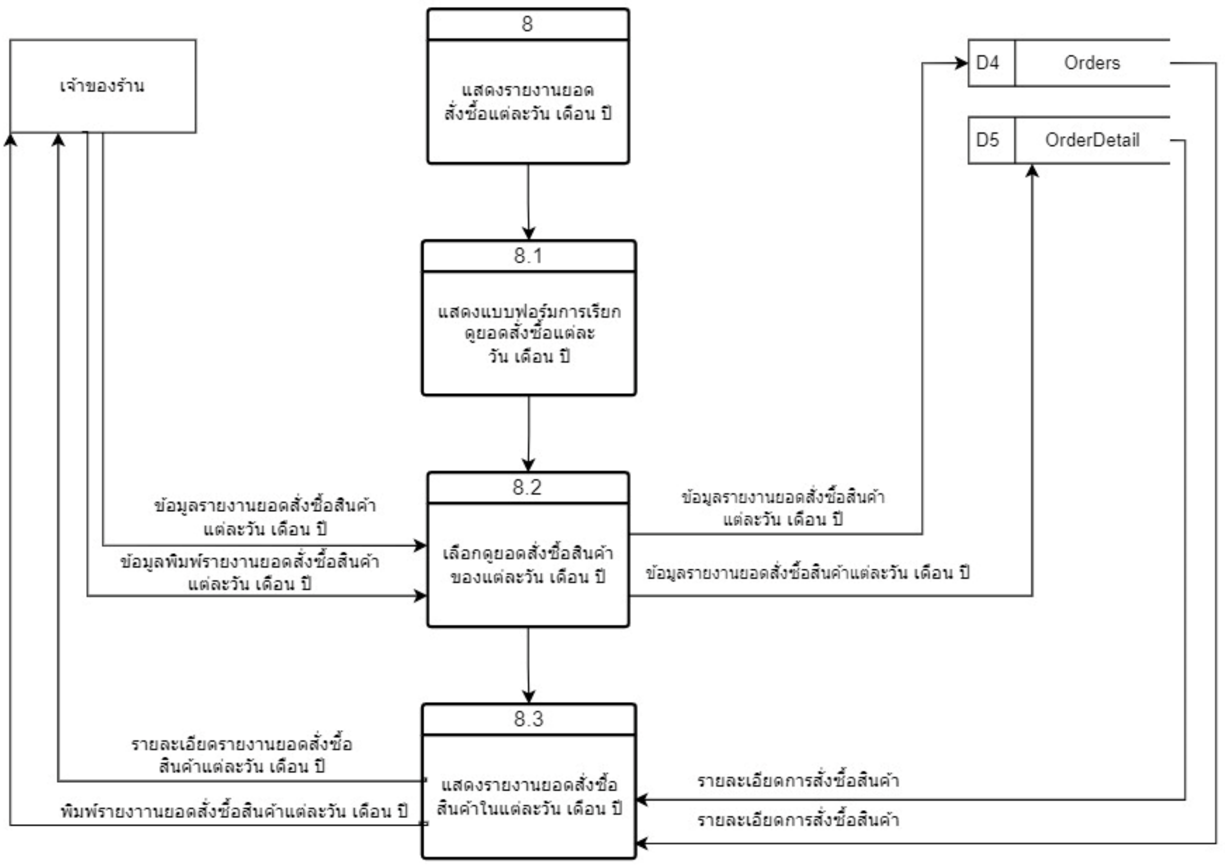
ภาพที่ 7.7 แผนภาพระดับที่ 1 ของกระบวนการที่ 5 จัดการสินค้า และระดับที่ 2 ของกระบวนการที่ 5.1 เพิ่มสินค้า



ภาพที่ 7.8 แผนภาพระดับที่ 1 ของกระบวนการที่ 6 ตรวจสอบการสั่งซื้อสินค้าของลูกค้า



ภาพที่ 7.9 แผนภาพระดับที่ 1 ของกระบวนการที่ 7 ตรวจสอบสินค้าคงเหลือ



ภาพที่ 7.10 แผนภาพระดับที่ 1 ของกระบวนการที่ 8 ตรวจสอบสินค้าคงเหลือ

คำอธิบายกระบวนการแผนภาพกระแสข้อมูล

คำอธิบายกระบวนการเชิงบรรยายของระบบขายหน้าร้าน POS ประกอบด้วยรายละเอียด ดังนี้

ตารางที่ 7.3 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 1 จัดการสมาชิก

System	**ระบบขายหน้าร้าน POS**
DFD Number	1
Process Name	จัดการสมาชิก
Input Data Flow	ข้อมูลสมาชิก
Output Data Flow	รายละเอียดสมาชิก
Data Stored Used	แฟ้มสมาชิก แฟ้มกำหนดสิทธิ์สมาชิก
Description	ระบบจัดการสมาชิกเพื่อเพิ่มข้อมูลสมาชิกเข้ามาในระบบใหม่ เช่น พนักงานขาย หรือเจ้าของร้าน โดยระบบจะสามารถกำหนดสิทธิ์เข้าใช้งานได้ว่าให้สิทธิ์เป็น พนักงานขาย หรือ เจ้าของร้าน ซึ่งสิทธิ์ในการใช้งานระบบแตกต่างกัน

ตารางที่ 7.4 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 2 เข้าสู่ระบบ

System	**ระบบขายหน้าร้าน POS**
DFD Number	2
Process Name	เข้าสู่ระบบ
Input Data Flow	ข้อมูล Username, Password
Output Data Flow	รายละเอียดการใช้งาน
Data Stored Used	แฟ้มกำหนดสิทธิ์สมาชิก
Description	ตรวจสอบสิทธิ์การเข้าใช้งานของสมาชิก รับค่า Username, Password เข้ามาตรวจสอบ โดยสิทธิ์เข้าใช้งานระบบมี 2 ระดับ ได้แก่ เจ้าของร้าน และพนักงานขาย

ตารางที่ 7.5 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 3 ค้นหาข้อมูลสินค้า

System	**ระบบขายหน้าร้าน POS**
DFD Number	3
Process Name	ค้นหาข้อมูลสินค้า
Input Data Flow	ข้อมูลรายการสินค้าที่ค้นหา
Output Data Flow	รายละเอียดรายการสินค้าที่ค้นหา
Data Stored Used	แฟ้มข้อมูลสินค้า
Description	ระบบจะรับค่ารหัสสินค้าที่ต้องการค้นหา โดยจะนำรหัสไปค้นหาจากแฟ้มข้อมูลสินค้าในระบบ และแสดงรายละเอียดสินค้ามาแสดงบนหน้าจอระบบ

ตารางที่ 7.6 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 4 สั่งซื้อสินค้า

System	**ระบบขายหน้าร้าน POS**
DFD Number	4
Process Name	สั่งซื้อสินค้า
Input Data Flow	ข้อมูลการสั่งซื้อสินค้า
Output Data Flow	รายละเอียดการสั่งซื้อสินค้า
Data Stored Used	แฟ้มข้อมูลสินค้า แฟ้มประเภทสินค้า แฟ้มสั่งซื้อสินค้า แฟ้มรายละเอียดการสั่งซื้อ
Description	พนักงานขายจะสแกนบาร์โค้ดที่รหัสสินค้าเพื่อนำสินค้าเข้าสู่รายการสั่งซื้อ จากนั้นระบบจะคำนวณราคาสินค้าทั้งหมด และแสดงรายละเอียดการสั่งซื้อ และพิมพ์ใบเสร็จรับเงินให้ลูกค้า

ตารางที่ 7.7 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 5 จัดการสินค้า

System	**ระบบขายหน้าร้าน POS**
DFD Number	5
Process Name	จัดการสินค้า
Input Data Flow	ข้อมูลรายการสินค้า
Output Data Flow	รายละเอียดการเพิ่มรายการสินค้า
Data Stored Used	แฟ้มข้อมูลสินค้า แฟ้มประเภทสินค้า
Description	เจ้าของร้านจะเพิ่มสินค้าใหม่เข้าสู่ระบบ โดยการกรอกข้อมูลรายการสินค้า เช่น ชื่อสินค้า ราคาสินค้า จำนวนสินค้า เป็นต้น จากนั้นระบบก็จะบันทึกข้อมูลสินค้าเข้าสู่ระบบ

ตารางที่ 7.8 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 6 ตรวจสอบการสั่งซื้อ

System	**ระบบขายหน้าร้าน POS**
DFD Number	6
Process Name	ตรวจสอบการสั่งซื้อของลูกค้า
Input Data Flow	ข้อมูลการตรวจสอบการสั่งซื้อของลูกค้า
Output Data Flow	รายละเอียดการตรวจสอบการสั่งซื้อของลูกค้า
Data Stored Used	แฟ้มสั่งซื้อสินค้า แฟ้มรายละเอียดการสั่งซื้อ
Description	เจ้าของร้านจะสามารถตรวจสอบการสั่งซื้อของลูกค้าได้ สามารถตรวจสอบได้ว่าสินค้ามีการสั่งซื้อกี่รายการได้

ตารางที่ 7.9 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 7 ตรวจสอบสินค้าคงเหลือ

System	ระบบขายหน้าร้าน POS
DFD Number	7
Process Name	ตรวจสอบสินค้าคงเหลือ
Input Data Flow	ข้อมูลการตรวจสอบสินค้าคงเหลือ
Output Data Flow	รายละเอียดการตรวจสอบสินค้าคงเหลือ
Data Stored Used	แฟ้มประเภทสินค้า แฟ้มสั่งซื้อสินค้า แฟ้มรายละเอียดการสั่งซื้อ
Description	ระบบจะสามารถสรุปจำนวนสินค้าที่คงเหลือได้ หรือสามารถเลือกดูจำนวนสินค้าคงเหลือแต่ละรายการสินค้าได้

ตารางที่ 7.10 คำอธิบายกระบวนการเชิงบรรยายของกระบวนการทำงานที่ 7 แสดงรายงานยอดสั่งซื้อแต่ละวัน เดือน ปี

System	ระบบขายหน้าร้าน POS
DFD Number	8
Process Name	แสดงรายงานยอดสั่งซื้อแต่ละวัน เดือน ปี
Input Data Flow	ข้อมูลรายงานยอดสั่งซื้อสินค้าแต่ละวัน เดือน ปี และ ข้อมูลพิมพ์รายงานยอดสั่งซื้อสินค้าแต่ละวัน เดือน ปี
Output Data Flow	รายละเอียดรายงานยอดสั่งซื้อสินค้าแต่ละวัน เดือน ปี และ
Data Stored Used	แฟ้มสั่งซื้อสินค้า แฟ้มรายละเอียดการสั่งซื้อ และ พิมพ์รายงานยอดสั่งซื้อสินค้าแต่ละวัน เดือน ปี
Description	ระบบสามารถที่จะสรุปรายการสั่งซื้อ และพิมพ์รายการสั่งซื้อสินค้าเป็นรายวัน รายเดือน รายปีได้

การสร้างแผนภาพ E-R

ขั้นตอนการสร้างแผนภาพ E-R ได้อธิบายไว้อย่างละเอียดในบทที่ 5 การออกแบบฐานข้อมูล ดังนั้นจึงสามารถอธิบายขั้นตอนการสร้างแผนภาพ E-R ดังนี้

1. การวิเคราะห์ข้อมูลเพื่อกำหนดเอนทิตีของระบบ

การวิเคราะห์ข้อมูลเพื่อกำหนดเอ็นทิตี้ของระบบจะเริ่มต้นพิจารณาจากแหล่งจัดเก็บข้อมูล (Data Store) ที่ได้สร้างไว้ในแผนภาพกระแสข้อมูล เนื่องจากได้ผ่านการวิเคราะห์ถึงความจำเป็นในการจัดเก็บข้อมูลดังกล่าวไว้ในระบบแล้ว ประกอบด้วย

1) แฟ้มสมาชิก

2) แฟ้มกำหนดสิทธิ์สมาชิก

3) แฟ้มสินค้า

4) แฟ้มประเภทสินค้า

5) แฟ้มการสั่งซื้อสินค้า

6) แฟ้มรายละเอียดการสั่งซื้อสินค้า

เนื่องจากแฟ้มข้อมูลได้ผ่านการวิเคราะห์จากขั้นตอนการสร้างแผนภาพกระแสข้อมูลแล้ว จึงสามารถที่จะนำไปสร้างเป็นเอ็นทิตี้ของระบบขายหน้าร้าน POS ได้ทั้งหมด ดังนี้

1) สมาชิก (User Detail)

2) สิทธิ์สมาชิก (User Role)

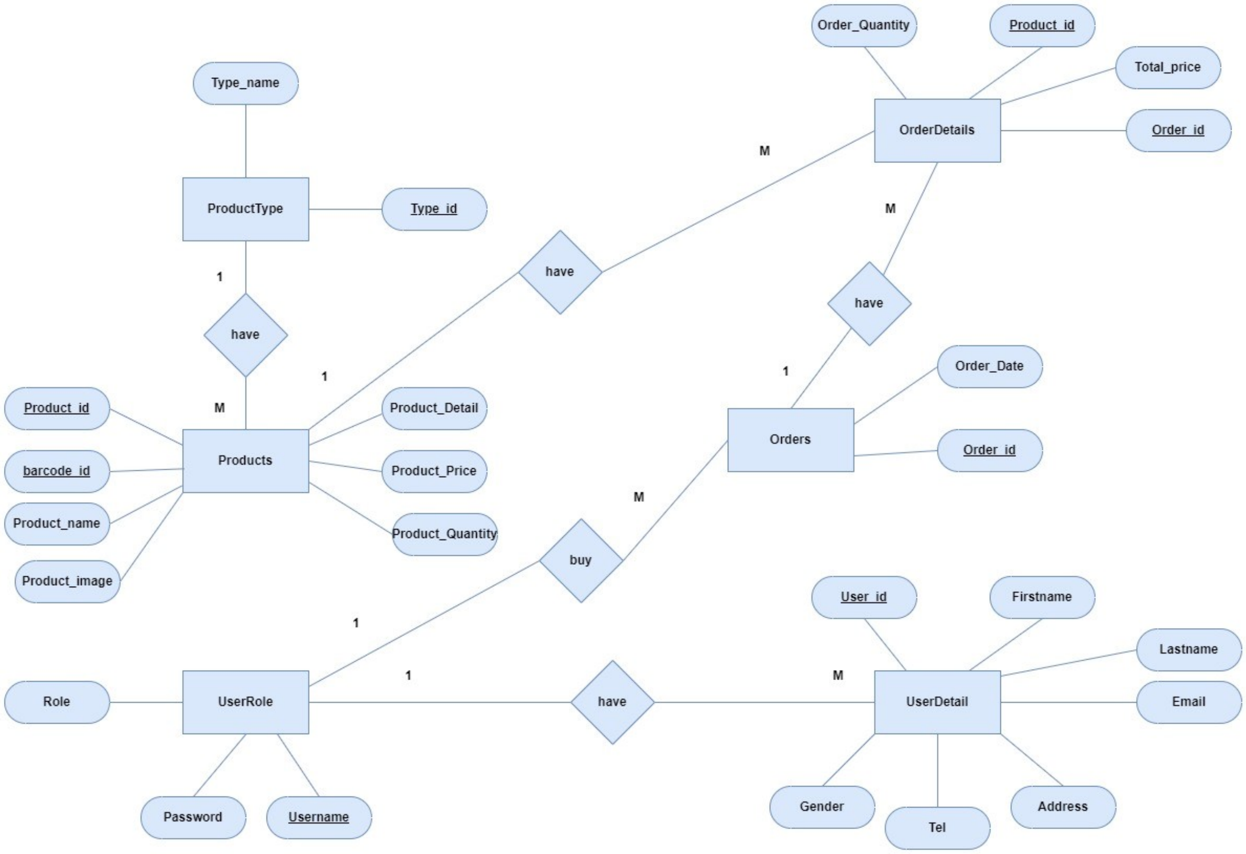
3) สินค้า (Products)

4) ประเภทสินค้า (Product Type)

5) การสั่งซื้อสินค้า (Orders)

6) แฟ้มรายละเอียดการสั่งซื้อสินค้า (Order Detail)

2. แผนภาพ E-R



ภาพที่ 7.11 แผนภาพ E-R ระบบขายหน้าร้าน POS

3. วิเคราะห์ความสัมพันธ์ระหว่างเอ็นทิตี้

การวิเคราะห์ความสัมพันธ์ระหว่างเอนทิตี จะพิจารณาจำนวนข้อมูลระหว่างเอนทิตีที่สัมพันธ์กันแต่ละคู่ไป จนกระทั่งครบทุกเอนทิตีที่กำหนดไว้ ซึ่งในที่นี้จะพิจารณาความสัมพันธ์ระหว่างเอนทิตีโดยเรียงลำดับตามความจำเป็นของการใช้งานกลุ่มข้อมูลต่างๆ จากภาพที่ 7.11 โดยประกอบด้วย

3.1 ความสัมพันธ์ระหว่างสมาชิก (User Detail) และเอนทิตีกำหนดสิทธิ์ของสมาชิก (User Role)

โดยที่ กำหนดเอนทิตีสิทธ์ของสมาชิก 1 สิทธิ์ มีสมาชิกในระบบได้มากกว่า 1 คน จึงแสดง “M” กำกับไว้หน้าเอนทิตีสมาชิก และ เอ็นทิตี้สมาชิก 1 คนสามารถมีสิทธิเข้าใช้ระบบได้เพียง 1 สิทธิ์ จึงแสดง “1” กำกับไว้หน้าเอ็นทิตี้กำหนดสิทธิ์ของสมาชิก

3.2 ความสัมพันธ์ระหว่างประเภทสินค้า (ProductType) และเอ็นทิตี้กำหนดสินค้า (Products)

โดยที่ กำหนดเอ็นทิตี้ประเภทสินค้า 1 ประเภท มีสินค้าได้มากกว่า 1 ชิ้น จึงแสดง “M” กำกับไว้หน้าเอ็นทิตี้สินค้า และ เอ็นทิตี้สินค้า 1 ชิ้น สามารถมีประเภทสินค้าได้เพียง 1 ประเภท จึงแสดง “1” กำกับไว้หน้าเอนทิตีกำหนดสิทธิ์ประเภทสินค้า

3.3 ความสัมพันธ์ระหว่างสิทธิ์เข้าใช้งานของสมาชิก (UserRole) และเอ็นทิตี้การสั่งซื้อสินค้า (Orders)

โดยที่ กำหนดเอ็นทิตี้สิทธิ์เข้าใช้งานของสมาชิก 1 สิทธ์ จะซื้อสินค้าได้มากกว่า 1 รายการ จึงแสดง “M” กำกับไว้หน้าเอ็นทิตี้การสั่งซื้อสินค้า และ เอนทิตีการสั่งซื้อสินค้า 1 รายการ จะมีสิทธิ์สมาชิกที่ซื้อได้เพียง 1 สิทธิ์ จึงแสดง “1” กำกับไว้หน้าเอนทิตีสิทธิ์เข้าใช้งานของสมาชิก

3.4 ความสัมพันธ์ระหว่างเอนทิตีการสั่งซื้อสินค้า (Orders) และเอนทิตีรายละเอียดการสั่งซื้อ (OrderDetail) โดยที่ กำหนดเอนทิตีการสั่งซื้อสินค้า 1 รายการ จะมีรายละเอียดการสั่งซื้อมากกว่า 1 รายการ จึงแสดง “M” กำกับไว้หน้าเอนทิตีรายละเอียดการสั่งซื้อ และ เอนทิตีรายละเอียดการสั่งซื้อ 1 รายการ จะมีการสั่งซื้อสินค้า 1 รายการ จึงแสดง “1” กำกับไว้หน้าเอนทิตีการสั่งซื้อสินค้า

3.5 ความสัมพันธ์ระหว่างเอนทิตีสินค้า (Products) และเอนทิตีรายละเอียดการสั่งซื้อ (OrderDetail)

โดยที่ เอ็นทิตี้สินค้า 1 ชิ้น จะมีรายละเอียดการสั่งซื้อได้มากกว่า 1 รายการ จึงแสดง “M” กำกับไว้หน้าเอ็นทิตี้รายละเอียดการสั่งซื้อ และ เอ็นทิตี้รายละเอียดการสั่งซื้อ 1 รายการ จะมีสินค้าที่อยู่ในรายละเอียดการการสั่งซื้อแต่ละครั้งได้เพียง 1 ชิ้น จึงแสดง “1” กำกับไว้หน้าเอ็นทิตี้สินค้า

โครงสร้างฐานข้อมูล

เมื่อได้พิจารณาตัวอย่างข้อมูลในรีเลชันต่าง ๆ แล้วว่ามีลักษณะของข้อมูลในรูปแบบใด ในขั้นตอนถัดไปจะเป็นการออกแบบโครงสร้างฐานข้อมูลของระบบการขายหน้าร้าน POS ดังนี้

ตารางที่ 7.11 กำหนดสิทธิ์เข้าใช้งานสมาชิก (UserRole)

Attribute	Description	Type	Key	Remark
Username	ชื่อเข้าใช้ระบบ	Varchar(20)	PK	
Password	รหัสผ่านเข้าใช้ระบบ	Varchar(20)		
Role	สิทธิ์เข้าใช้ระบบ	ENUM (เจ้าของร้าน,พนักงานขาย)		

ตารางที่ 7.12 สมาชิก (UserDetail)

Attribute	Description	Type	Key	Remark
User_id	รหัสสมาชิก	Varchar(10)	PK	
Username	ชื่อเข้าใช้ระบบ	Varchar(20)	FK	
Firstname	ชื่อ	Varchar(20)		
Lastname	สกุล	Varchar(20)		
Email	อีเมล	Varchar(30)		
Address	ที่อยู่	Text		
Tel	เบอร์โทร	Varchar(20)		
Gender	เพศ	Varchar(10)		

ตารางที่ 7.13 ประเภทสินค้า (ProductType)

Attribute	Description	Type	Key	Remark
Type_id	รหัสประเภทสินค้า	Varchar(10)	PK	
Type_name	ชื่อประเภทสินค้า	Varchar(50)		

ตารางที่ 7.14 สินค้า (Products)

Attribute	Description	Type	Key	Remark
Product_id	รหัสสินค้า	Varchar(10)	PK	
Barcode_id	รหัสบาร์โค๊ดสินค้า	Varchar(10)		
Type_id	รหัสประเภทสินค้า	Varchar(10)	FK	
Prodcut_name	ชื่อสินค้า	Varchar(50)		
Product_image	รูปภาพสินค้า	Image		
Product_detail	รายละเอียดสินค้า	Text		
Project_Price	ราคาสินค้า	Double		
Product_Quantity	จำนวนสินค้าคงเหลือ	Int(5)		

ตารางที่ 7.15 การสั่งซื้อสินค้า (Orders)

Attribute	Description	Type	Key	Remark
Order_id	รหัสสั่งซื้อสินค้า	Varchar(10)	PK	
Order_date	วันที่สั่งซื้อสินค้า	Datetime		

ตารางที่ 7.16 รายละเอียดการสั่งซื้อสินค้า (OrderDetail)

Attribute	Description	Type	Key	Remark
Product_id	รหัสสินค้า	Varchar(10)	PK, FK	
Order_id	รหัสสั่งซื้อสินค้า	Varchar(10)	PK, FK	
Total_price	ราคารวม	Double		
Order_Quantity	จำนวนสั่งซื้อสินค้าต่อรายการ	Int(5)		

สรุปท้ายบทที่ 7

เนื้อหาในบทนี้จะยกตัวอย่างการประยุกต์ใช้งานจริงของการวิเคราะห์และออกแบบระบบ กรณีศึกษาระบบขายหน้าร้าน (Point of Sale System : POS) เป็นระบบที่ใช้ในธุรกิจขายปลีกและบริการ สำหรับลูกค้าที่ต้องการการชำระเงินตามร้านหรือจุดขายหน้าร้าน ในบทนี้ได้ยกตัวอย่างตั้งแต่ขั้นตอนแรกไปจนถึงการออกแบบฐานข้อมูลของระบบขายหน้าร้าน POS มีการยกตัวอย่างอธิบายแต่ละขั้นตอนอย่างละเอียด โดยมีหัวข้อดังนี้

1. การรวบรวมความต้องการใช้งานของผู้ใช้งาน
2. การวิเคราะห์ความต้องการของระบบ
3. การสร้างแผนภาพกระแสข้อมูล
4. คำอธิบายกระบวนการแผนภาพกระแสข้อมูล
5. การสร้างแผนภาพ E-R
6. โครงสร้างฐานข้อมูล

คำถามท้ายบทที่ 7

1. จากกรณีศึกษาระบบขายหน้าร้าน POS ให้วิเคราะห์ความต้องการของระบบเพิ่มขึ้นอย่างน้อย 3 ฟังก์ชัน
2. ฟังก์ชันที่เพิ่มเข้ามาใหม่ให้เพิ่มกระบวนการในแผนภาพระดับที่ 0
3. ฟังก์ชันที่เพิ่มเข้ามาใหม่ให้เพิ่มเอนทิตีในแผนภาพ E-R
4. ฟังก์ชันที่เพิ่มข้ามาใหม่ให้เพิ่มตารางในโครงสร้างฐานข้อมูล
5. ให้ออกแบบต้นแบบ (Prototype) ส่วนต่อประสานกับผู้ใช้ (User Interface) หน้าแสดงรายการสินค้า หน้าสั่งซื้อสินค้า และหน้าแสดงรายละเอียดการชำระเงิน

บรรณานุกรม

ณัฐพล ใยไพโรจน์. (2561). *Digital Marketing5G : concept & case study*. กรุงเทพฯ: ไอดีซี พรีเมียร์.

นฤมล ญาณสมบัติ และเมธารัตน์ จันตะนี. (2562). การจัดการการค้าสมัยใหม่ในยุคดิจิทัล. *วารสารวิทยาการจัดการปริทัศน์*, 21(2), 211-220.

นาฎวราพร สิมทอง. (2552). *การวิเคราะห์และออกแบบระบบสารสนเทศงานวิจัย กรณีศึกษา : คณะแพทยศาสตร์ มหาวิทยาลัยธรรมศาสตร์* (วิทยานิพนธ์). กรุงเทพฯ: มหาวิทยาลัยธุรกิจบัณฑิตย์.

ลั่นทม จอนจวบทรง และณธภร ธรรมบุญวริศ. (2562). การแปลงรูปสู่ธุรกิจดิจิทัลก่อนถูกยุบสลาย. *วารสารธุรกิจปริทัศน์*, 11(2), 216-231.

ศรีไพร ศักดิ์รุ่งพงศากุล และเจษฎพร ยุทธนวิบูลย์ชัย. (2549). *ระบบสารสนเทศและเทคโนโลยีการจัดการความรู้* (พิมพ์ครั้งที่ 10). กรุงเทพฯ: ซีเอ็ดยูเคชั่น.

สำนักงานนวัตกรรมแห่งชาติ(องค์การมหาชน). (2562). *Innovation for the Future นวัตกรรมเพื่ออนาคต*. สืบค้นวันที่ 7 สิงหาคม 2566 จาก https://www.nia.or.th/frontend/bookshelf/zWf7zwTMcEWhI/62d3a82b0f129.pdf

สำนักงานราชบัณฑิตยสภา. (ม.ป.ป.). *พจนานุกรมฉบับราชบัณฑิตยสถาน พ.ศ. 2557*. สืบค้นวันที่ 7 สิงหาคม 2566 จาก https://dictionary.orst.go.th/

อรยา ปรีชาพานิช. (2557). *คู่มือเรียน การวิเคราะห์และออกแบบระบบ (System Analysis and Design) ฉบับสมบูรณ์*. กรุงเทพฯ: ไอดีซี พรีเมียร์.

โอภาส เอี่ยมสิริวงศ์. (2556). *การวิเคราะห์และออกแบบระบบ (System Analysis and Design) ฉบับปรับปรุงเพิ่มเติม*. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.

โอภาส เอี่ยมสิริวงศ์. (2558). *ระบบฐานข้อมูล (Database Systems) ฉบับปรับปรุงเพิ่มเติม*. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.

โอภาส เอี่ยมสิริวงศ์. (2561). *พาณิชย์อิเล็กทรอนิกส์ (มุมมองด้านการบริหาร)*. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.

Beck, K. & Andres, C. (2004). *Extreme Programming Explained: Embrace Change (The XP Series)* (2nd ed.). United States: Addison-Wesley.

Burghardt, A., Szybicki, D., Gierlak, P., Kurc, K., Pietruś, P., & Cygan, R. (2020). Programming of

Industrial Robots Using Virtual Reality and Digital Twins. *Applied Sciences*, 10(2), 486. MDPI AG. Retrieved from http://dx.doi.org/10.3390/app10020486

Connolly, T. M., & Begg, C. E. (2014). *Database Systems: A Practical Approach to Design, Implementation, and Management* (6th ed.). London, United Kingdom: Pearson.

Coronel, C., Morris, S., & Rob, P. (2017). *Database systems: Design, implementation, and management* (4th ed.). USA: Cengage Learning.

Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems* (7th ed.). London, United Kingdom: Pearson.

Kaewman, S., & Akararungruangkul, R. (2018). Heuristics Algorithms for a Heterogeneous Fleets VRP with Excessive Demand for the Vehicle at the Pickup Points, and the Longest Traveling Time Constraint: A Case Study in Prasitsuksa Songkloe, Ubonratchathani Thailand. *Logistics*, 2(3), 15. MDPI AG. Retrieved from http://dx.doi.org/10.3390/logistics2030015

Parker, G.G., Van Alstyne, M.W., and Choudary, S.P. (2016). *Platform Revolution: How Networked Markets Are Transforming the Economy - and How to Make Them Work for You*. New York, United States: W.W. Norton & Company.

Preece, J., Rogers, Y., & Sharp, H. (2019). *Interaction design: Beyond human-computer interaction* (5th ed.). New York, United States: Wiley.

Satell, G. (2017). *Mapping innovation: a playbook for navigating a disruptive age*. United States: McGraw Hill.

Schwaber, K., & Beedle, M. (2001). *Agile software development with Scrum*. Prentice Hall PTR.

Shneiderman, B., & Plaisant, C. (2016). *Designing the user interface: Strategies for effective human-computer interaction* (6th ed). Pearson.

Svensson, H. and Host, M. (2005). Introducing an agile process in a software maintenance and evolution organization. *Ninth European Conference on Software Maintenance and Reengineering*. Manchester, UK, 2005, pp. 256-264. doi: 10.1109/CSMR.2005.33.

Svensson, H., & Höst, M. (2005). Views from an organization on how agile development affects its collaboration with a software development team. Lecture Notes in Computer

Science, 3547, 487–501. https://doi.org/10.1007/11497455_38

Theeraviriya, C., Pitakaso, R., Sillapasa, K., & Kaewman, S. (2019). Location Decision Making and Transportation Route Planning Considering Fuel Consumption. *Journal of Open Innovation: Technology, Market, and Complexity*, 5(2), 27. Elsevier BV. Retrieved from http://dx.doi.org/10.3390/joitmc5020027

Yablonski, J. (2020). *Laws of UX: Using Psychology to Design Better Products & Services (1st ed.)*. O'Reilly Media.

ดัชนี

A

Adapted Waterfall Model 30
Adaptive Software Development.................51
Agile Software Development........................42
Attributes..119

C

Candidate Key 116
Cardinality 115
CASE Tools 57
Command Language Interaction 143

D

Data 105
Data Flow Diagram 25, 34, 85, 95
Data Flows 87
Data Model 106
Data Query Language 113
Data Stores 90
Database Administrator...............................106
Database Designers 114
Database Management System103, 105, 112, 113
Database Model..107
Decision Support System................................. 6
Degree 115
Document Review.. 67
Domain 10, 115, 127
Dynamic System Development Method:... 52

E

E-business 14
Electronic Business 14
End-Users.. 114
Entity.. 106, 117
External Entities.. 88
Extreme Programming................................... 47

F

Feasibility Study 24, 53
Foreign Key... 117
Form Interaction... 146
Functional Requirement............................... 78

H

Hardware... 105

I

Incremental Model.. 30
Information System 1, 5, 7
Innovation..9
Innovation Matrix 9

Iteration 52

J

Jakob's Principle of Design 141

L

Logical Model... 25, 26

M

many-to-many relationship.........................108
Many-to-Many Relationships.......................123
Menu Interaction 144
Model Management... 6

N

Natural Language Interaction......................147
Network Database Model 108
Nonfunctional Requirement..........................79

O

Object-Based Interaction 146
Object-Oriented Database Model..............110
Observation 76
Office Automation System 7
One-to-Many Relationships.........................122
One-to-One Relationships 122

P

Primary Key..116
Problem Definition...................................10, 24
Procedure 113
Processes 86, 87
Program Flowchart...................................26, 34
Prototype.. 26

Q

Questionnaire.. 73

R

Rapid Prototyping Model.............................. 32
Real-Time Link.. 92
Relation... 115
Relational Database..................................... 114
Relational Model.. 100
Relationship 106, 122

S

Scrum 48
Software.. 105
Software Development Life Cycle 22
Spiral Model.. 31
Strong Entity.. 119
Structured Query Language 109, 125
Super key... 116
System Analysis.....................................25, 114
System Boundary..2
System Development Life Cycle 28, 63
System Flowchart 26

System Requirement 25
System Requirement Analysis 78
System Requirement Specification 25, 80
System Requirements 24

T

Table Relationship... 100
Transaction Processing System 4
Tuple 115

U

User Interface 140, 143
User Interface Design 135
User Interface Management System 138
User Requirement 25
User Requirement Gathering 64, 66

W

Waterfall Model 28
Weak Entity 119

ก

กระบวนการวิเคราะห์ความต้องการของระบบ 78
กระแสข้อมูล 87
การกำหนดความต้องการของระบบ 24
การกำหนดปัญหา 24
การจัดการแบบจำลอง 6
การโต้ตอบด้วยการทำงานเชิงวัตถุ 146
การโต้ตอบด้วยคำสั่ง 143
การโต้ตอบด้วยแบบฟอร์ม 146
การโต้ตอบด้วยภาษามนุษย์ 147
การโต้ตอบด้วยเมนูคำสั่ง 144
การพัฒนาซอฟต์แวร์แบบปรับตัว 51
การพัฒนาซอฟต์แวร์แบบเอจายล์ 42
การพัฒนาระบบแบบไม่หยุดนิ่ง 52
การพัฒนาระบบสารสนเทศ 22
การรวบรวมความต้องการของผู้ใช้งาน 64
การวงรอบ 52
การศึกษาความเป็นไปได้ 24
การสังเกตการณ์ 76
การสัมภาษณ์ 68
การออกแบบส่วนติดต่อผู้ใช้ 135

ข

ข้อกำหนดเกี่ยวกับความต้องการของระบบ ... 25, 80
ขอบเขตของระบบ 2
ข้อมูล 105

ค

ความต้องการของระบบ 25
ความต้องการที่เป็นฟังก์ชัน 78
ความต้องการที่ไม่เป็นฟังก์ชัน 79
ความสัมพันธ์ 122
ความสัมพันธ์แบบกลุ่มต่อกลุ่ม 123
ความสัมพันธ์แบบหนึ่งต่อกลุ่ม 122
ความสัมพันธ์แบบหนึ่งต่อหนึ่ง 122
ความสัมพันธ์แบบหลายต่อหลาย 108
ความสัมพันธ์ระหว่างตาราง 100

คาร์ดินัลลิตี้ 115
คีย์คู่แข่ง 116
คีย์นอก 117
คีย์หลัก 116

ซ

ซอฟต์แวร์ 105
ซูเปอร์คีย์ 116

ฐ

ฐานข้อมูลเชิงสัมพันธ์ 114

ด

ดาตาโฟล์ว 87
ดาตาสโตร์.. 90
ดีกรี.. 115
โดเมน 115, 127

ท

ทูเพิล 115

ธ

ธุรกิจดิจิทัล 14
ธุรกิจอิเล็กทรอนิกส์ 14

น

นวัตกรรม... 9
นักวิเคราะห์ระบบ 114
นักออกแบบฐานข้อมูล 114

บ

แบบจำลองการสร้างโปรแกรมต้นแบบในระยะ
เริ่มต้น 32
แบบจำลองข้อมูล 106
แบบจำลองเชิงตรรกะ 25
แบบจำลองฐานข้อมูล 107
แบบจำลองน้ำตก.. 28
แบบจำลองแบบเกลียว 31
แบบจำลองแบบเพิ่มผลลัพธ์ 30
แบบจำลองแบบสัมพันธ์ 100
แบบจำลองปรับปรุงน้ำตก 30

ป

โปรแกรมต้นแบบ 26
โปรเซส 85, 87, 91

ผ

ผังงานโปรแกรม 26
ผังงานระบบ 26
ผู้ใช้งาน.. 4, 10, 15
ผู้บริหารฐานข้อมูล... 106
แผนภาพกระแสข้อมูล 85, 95

พ

โพรซีเยอร์... 113

ร

ระบบจัดการฐานข้อมูล...................................... 112
ระบบจัดการสื่อประสานกับผู้ใช้......................... 138

ระบบฐานข้อมูล 105
ระบบบริหารจัดการฐานข้อมูล 103
ระบบประมวลสารสนเทศเชิงรายการ 4
ระบบสำนักงานอัตโนมัติ 7
ระบบสารสนเทศ 1
ระบบสารสนเทศเพื่อการจัดการ 5
ระบบสารสนเทศเพื่อสนับสนุนการตัดสินใจ 6
ระบบสารสนเทศสำหรับผู้บริหาร 7
ระเบียบวิธีการพัฒนาระบบ 33
รีเลชั่น 100, 115
เรียลไทม์ลิงก์ 92

ว

วงจรการพัฒนาระบบ 22

ห

หลักการออกแบบของจาคอป 141
แหล่งเก็บข้อมูล 90

อ

เอ็กซ์เทอร์นัลเอนทิตี 88
แอตทริบิวต์ 115

ฮ

ฮาร์ดแวร์ 105