



Fundamental of SOFTWARE ENGINEERING & Digital Transformation



Software Engineering :
An Important Role in
Digital Transformation

อธิบายกระบวนการ Digital Transformation
ด้วย Software Engineering ถ่ายทอดจากประสบการณ์
ในการพัฒนาระบบเทคโนโลยีดิจิทัลให้กับหน่วยงาน
ทั้งภาครัฐและเอกชน

ผู้แต่ง สร. ดร.ปานใจ สารภักตวงศ์
บรรณาธิการ ภีรพล คชาเจริญ



มีเพียง “ความรู้” เท่านั้นที่มนุษย์ใช้พลิก “โลก”
และเปลี่ยน “ชีวิต” เราจึงสร้างสรรค์ และส่งมอบ “ความรู้”
ในรูปแบบที่ดีกว่า เพื่อให้คนไทย “เรียนรู้” ได้ตลอดชีวิต

Only “Knowledge” can help human
change “The World” and “Their Lives”.

With this truth, it drives us to deliver
“Knowledge” for Thai being able to
“Learn” better everyday.



Fundamental of Software Engineering & Digital Transformation

Writer	รองศาสตราจารย์ ดร.ปานใจ ชารัทศนวงศ์
Editor	กิริพล คุชาเจริญ
Graphic Designers	ชวนันท์ รัตนะ, กมลชนก ชัยสาครสมุทร
Page Layout	ถมทวานิตย์ มนตรีมโนรมย์
Proofreader	เกษรา พรพัฒน์มงคล
Publishing Coordinators	วรพล ณธิกุล, สุพัตรา อาจปรุ, ศรัณย์ คมขำ

เครื่องหมายการค้าอื่นๆ ที่อ้างถึงเป็นของบริษัทนั้นๆ สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดยบริษัท ไอดีซี พรีเมียร์ จำกัด ห้ามลอกเลียนไม่ว่า ส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ ไม่ว่าในรูปแบบใดๆ นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้จัดพิมพ์เท่านั้น

บริษัท ไอดีซี พรีเมียร์ จำกัด จัดตั้งขึ้นเพื่อเผยแพร่ความรู้ที่มีคุณภาพสู่ผู้อ่านชาวไทย เรายินดีรับงานเขียนของนักวิชาการและนักเขียนทุกท่าน ท่านผู้สนใจกรุณาติดต่อผ่านทางอีเมลที่ infopress@idcpremier.com หรือทางโทรศัพท์หมายเลข 0-2962-1081 (อัตรานาที 10 คู่สาย) โทรสาร 0-2962-1084

สร้างสรรค์โดย



พิมพ์และจัดจำหน่ายโดย



พิมพ์ครั้งที่ 1 มีนาคม 2565

ข้อมูลทางบรรณานุกรม :

รองศาสตราจารย์ ดร.ปานใจ ชารัทศนวงศ์
Fundamental of Software Engineering &
Digital Transformation
นนทบุรี : ไอดีซีฯ, 2565
376 หน้า

1. วิศวกรรมซอฟต์แวร์
I ชื่อเรื่อง

005.1

Barcode 885-916-100-987-0

ราคา 450 บาท

จัดพิมพ์และจัดจำหน่ายโดย

บริษัท ไอดีซี พรีเมียร์ จำกัด
200 หมู่ 4 ชั้น 19 ห้อง 1901
จัสมินอินเตอร์เนชั่นแนลทาวเวอร์
ถ.แจ้งวัฒนะ อ.ปากเกร็ด จ.นนทบุรี 11120
โทรศัพท์ 0-2962-1081 (อัตรานาที 10 คู่สาย)
โทรสาร 0-2962-1084

ร้านค้าและตัวแทนจำหน่าย

โทรศัพท์ 0-2962-1081-3 ต่อ 112-114
โทรสาร 0-2962-1084

สมาชิกสัมพันธ์

โทรศัพท์ 0-2962-1081-3 ต่อ 121
โทรสาร 0-2962-1084

PREFACE

หนังสือ Fundamental of Software Engineering & Digital Transformation เล่มนี้ได้รับการสนับสนุนจากโครงการตำรา คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร เป็นหนังสือที่ผู้เขียนตั้งใจรวบรวมและประยุกต์หลักทฤษฎีวิศวกรรมซอฟต์แวร์ ซึ่งเป็นเครื่องมือสำคัญในการที่องค์กรจะปรับเปลี่ยนสู่ดิจิทัล (Digital Transformation) ผสมผสานกับประสบการณ์ของผู้เขียนที่พัฒนาระบบเทคโนโลยีดิจิทัลให้กับหน่วยงานทั้งภาครัฐและเอกชน ตั้งแต่ระดับพัฒนาซอฟต์แวร์ประยุกต์ จนถึงการทำแผนพัฒนาเทคโนโลยีสารสนเทศและการสื่อสาร รวมทั้งแผนพัฒนาดิจิทัล ซึ่งได้พบว่าการที่องค์กรจะปรับเปลี่ยนการทำงานเป็นดิจิทัลนั้น ต้องอาศัยความรู้และความเข้าใจในหลายมิติ

ดังนั้น ผู้เขียนจึงเกิดแรงบันดาลใจที่จะเขียนหนังสือที่อธิบายหลักการวิศวกรรมซอฟต์แวร์ให้เข้าใจได้โดยง่าย และอธิบายถึงเทคโนโลยีต่างๆ เทคนิคต่างๆ ที่ใช้เป็นเครื่องมือในการปรับเปลี่ยนให้เป็นองค์กรดิจิทัล นอกจากนี้ยังได้รวบรวมแนวคิดในการพัฒนาซอฟต์แวร์แบบใหม่ๆ และกรณีศึกษา เพื่อให้ผู้อ่านได้ก้าวทันตามการเปลี่ยนแปลงของเทคโนโลยีด้านวิศวกรรมซอฟต์แวร์

หนังสือเล่มนี้จะอธิบายกระบวนการการปรับเปลี่ยนสู่ดิจิทัลด้วยวิศวกรรมซอฟต์แวร์ โดยครอบคลุมเนื้อหา ดังนี้ บทที่ 1 อธิบายภาพรวมของการปรับเปลี่ยนสู่ดิจิทัล บทที่ 2 การวิเคราะห์และออกแบบเชิงโครงสร้าง บทที่ 3 การวิเคราะห์และออกแบบเชิงวัตถุ บทที่ 4 UML ภาษาการสร้างแบบจำลองแบบครบวงจร บทที่ 5 กระบวนการซอฟต์แวร์ บทที่ 6 การจัดการเทคโนโลยีดิจิทัล บทที่ 7 วิศวกรรมความต้องการ บทที่ 8 การออกแบบระบบ บทที่ 9 การทำให้เกิดผลและการบำรุงรักษาระบบ บทที่ 10 วิศวกรรมซอฟต์แวร์เชิงเซอร์วิส และบทที่ 11 การพัฒนาบุคลากรด้านดิจิทัล

หนังสือเล่มนี้จึงเหมาะสำหรับผู้บริหารเทคโนโลยีสารสนเทศระดับสูง (CIO) ผู้บริหารโครงการด้านเทคโนโลยีสารสนเทศ วิศวกรซอฟต์แวร์ นักวิเคราะห์และออกแบบระบบ นักพัฒนาซอฟต์แวร์ โปรแกรมเมอร์ นิสิตนักศึกษาที่เรียนด้านวิทยาการคอมพิวเตอร์ และเทคโนโลยีสารสนเทศ และผู้สนใจด้านการพัฒนาซอฟต์แวร์

ผู้เขียนหวังเป็นอย่างยิ่งว่า หนังสือเล่มนี้จะเป็นประโยชน์ต่อการพัฒนาองค์กรดิจิทัล หรือได้มีส่วนร่วมที่ทำให้การพัฒนาซอฟต์แวร์ของประเทศให้ประสบผลสำเร็จ และขอขอบคุณคณะวิทยาศาสตร์ ม.ศิลปากร ที่สนับสนุนทุนในการจัดทำหนังสือนี้

รองศาสตราจารย์ ดร.ปานใจ ชารัตตนวงค์

อาจารย์ประจำภาควิชาคอมพิวเตอร์

คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร

การศึกษา

- ปริญญาตรี มหาวิทยาลัยมหิดล
- ปริญญาโท (วิทยาการคอมพิวเตอร์) จุฬาลงกรณ์มหาวิทยาลัย
- ปริญญาเอก Doctor of Technical Science (Computer Science) Asian Institute of Technology (AIT)

ตำแหน่งปัจจุบัน

- อาจารย์ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร

ผลงานวิจัย (บางส่วน)

- โครงการการพัฒนาหุ่นยนต์แพทย์อัจฉริยะสำหรับการวินิจฉัยและรักษาผ่านระบบโทรเวช เพื่อลดภาระรายจ่ายของประเทศ ระยะที่ 1, 2 โครงการพัฒนาระบบการประมวลผลข้อมูลทางการศึกษาด้วยหลักการข้อมูลขนาดใหญ่ (Big Data) และโอเพ่นโฟลว์ (Open Flow) โครงการพัฒนาระบบการเรียนรู้ทางการแพทย์ผ่านระบบเทคโนโลยีเพื่อการศึกษา
- โครงการ Real-Time Haze Monitoring and Forest Fire Detection Information Centric Networks (SEA HAZEMON@TEIN)
- โครงการพัฒนาและถ่ายทอดองค์ความรู้ระบบไฟระวังไฟป่าอัจฉริยะ

ผลงานอื่น (บางส่วน)

- ที่ปรึกษาการจัดทำแผนพัฒนาดิจิทัลเพื่อเศรษฐกิจการพาณิชย์ กระทรวงพาณิชย์
- ที่ปรึกษาการจัดทำแผนพัฒนาดิจิทัลเพื่อการทดสอบทางการศึกษาแห่งชาติ สถาบันทดสอบทางการศึกษาแห่งชาติ (องค์การมหาชน) กระทรวงศึกษาธิการ

EDITOR'S NOTE

ในช่วง 2 ปีที่ผ่านมา การระบาดใหญ่ของ COVID-19 ก่อให้เกิดบทเรียนที่สำคัญคือ การปรับเปลี่ยนเพื่อเปลี่ยนแปลง (Transformative Change) อย่างแท้จริงนั้นไม่ยากอย่างที่คิด หากมีแรงจูงใจมากพอ! โดยในสถานการณ์เช่นนี้เป็นการเปลี่ยนแปลงเพื่อพยายามเอาชีวิตรอดในโลกที่เปลี่ยนไปอย่างรวดเร็วฉับพลัน จนทำให้นึกถึงคำว่า “Change or Die”

เทรนด์เทคโนโลยีที่อาจเข้ามาเปลี่ยนวิถีของธุรกิจกระแสหลักให้กลายเป็น Digital Ecosystem ในอุตสาหกรรมต่างๆ เช่น Everything-as-a-Service ที่ AI Solutions สามารถให้บริการครบทุกสิ่งตั้งแต่ Marketing ไปจนถึง HR, No-Code จะได้รับความนิยมโดยความรู้ด้านโปรแกรมมิ่งจะไม่ใช่อุปสรรคอีกต่อไป, Virtualization การสัมผัสกับประสบการณ์เสมือนจริงตามแนวคิด “Metaverse” ที่เสนอโดย Mark Zuckerberg ผู้ก่อตั้ง Facebook, Cloud Solutions ที่ช่วยประหยัดต้นทุนไอที เช่น จ่ายเท่าที่ใช้ ยืดหยุ่นได้ตามขนาดธุรกิจ, IoT (Internet of Things) การเก็บข้อมูลผ่านอุปกรณ์ Wearable เช่น ธุรกิจ Insurance เสนอประกันภัยรถยนต์แบบ Personalized Product เช่น Pay per Use หรือการทำ Precision Farming ที่นำข้อมูลจาก Sensors มาวิเคราะห์เพื่อสร้างโมเดลการเพาะปลูกที่แม่นยำ, Automation ระบบอัตโนมัติที่ทำงานซ้ำซากแทนมนุษย์, Blockchain ระบบการจัดเก็บข้อมูลที่เชื่อมโยงโครงข่ายของผู้ใช้โดยไม่ผ่านคนกลาง เช่น Fintech, Cryptocurrency เป็นต้น ซึ่งเทคโนโลยีเหล่านี้จะเข้ามาเปลี่ยนวิถีของธุรกิจกระแสหลักที่อาจทำให้ธุรกิจจำเป็นต้องลุกขึ้นมา Innovate & Transform Business กันอีกครั้ง

มุมมองด้าน Software Engineering, Digital Transformation การปรับเปลี่ยนสู่ดิจิทัลนั้นทำให้วิธีการพัฒนาซอฟต์แวร์เปลี่ยนไป แนวโน้มในปัจจุบันจะเน้นการพัฒนาแอปพลิเคชันบน Mobile Devices, IoT และ Platform เป็นต้น ฉะนั้น วิศวกรรมซอฟต์แวร์จำเป็นต้องตอบสนองอย่างเหมาะสม เพื่อให้โครงการซอฟต์แวร์นั้นประสบความสำเร็จในโลกดิจิทัล รวมถึงการมี Team Culture ที่เหมาะสมสำหรับทีมนักพัฒนาซอฟต์แวร์ เพื่อเตรียมความพร้อมสำหรับโครงการที่สามารถแข่งขันได้ในอนาคต

มุมมองด้าน Business, Digital Transformation เป็นแนวคิดในการสร้างโมเดลธุรกิจใหม่ หรือปรับเปลี่ยนโมเดลธุรกิจเดิม โดยอิงจากการปรับเปลี่ยนสู่ดิจิทัลที่เพิ่มขึ้นในสังคม ธุรกิจต้องตอบสนองต่อการเปลี่ยนแปลงไปสู่ระบบนิเวศดิจิทัลที่เพิ่มขึ้น และต้องปรับตัวตามการเปลี่ยนแปลงให้ทัน ต้องมองหาโซลูชันเทคโนโลยีดิจิทัลมาช่วยแก้ปัญหาและปรับปรุงการดำเนินงานองค์กรธุรกิจ เช่น การใช้ Cloud Computing เพื่อลดฮาร์ดแวร์ การสร้างระบบ Automation เพื่อเปลี่ยนกระบวนการทำงานใหม่ การพัฒนา Technology Integration เพื่อประยุกต์ใช้เทคโนโลยีต่างๆ ร่วมกัน และการใช้ Social Network/Media ในการทำงานและดำเนินกิจกรรมทางธุรกิจ เป็นต้น

ฉะนั้น เมื่อเรากล่าวถึง Digital Transformation จึงต้องเอ่ยถึงวิศวกรรมซอฟต์แวร์ (Software Engineering) อย่างหลีกเลี่ยงไม่ได้ เพราะมีหน้าที่นำแนวคิดของเทคโนโลยียุคใหม่มาสู่ชีวิตและใช้งานได้จริง และสร้างการเติบโตทางธุรกิจ โดยในโลกปัจจุบัน วิศวกรรมซอฟต์แวร์ที่มีบทบาทอย่างมากในการทำ Digital Transformation และการทำ Digitization/Digitalization ก็เป็นความท้าทายสำหรับองค์กรในการดำเนินโครงการต่างๆ ให้ประสบความสำเร็จ

ภีรพล คชาเจริญ

บรรณาธิการ

CONTENT

CHAPTER

1

การปรับเปลี่ยนสู่ดิจิทัล DIGITAL TRANSFORMATION

1

- 1.1 เศรษฐกิจดิจิทัล (Digital Economy) 2
- 1.2 ความหมายของเทคโนโลยีดิจิทัล (Digital Technology Definition) 3
- 1.3 องค์กรดิจิทัล (Digital Enterprises) 5
- 1.4 การปรับเปลี่ยนสู่ดิจิทัล (Digital Transformation) 6
 - 1.4.1 ขั้นตอนการปรับเปลี่ยนสู่ดิจิทัล (Digital Transformation Roadmap) 6
 - 1.4.2 กรอบการปรับเปลี่ยนสู่ดิจิทัล (Digital Transformation Framework) 8
- 1.5 บทบาทของวิศวกรซอฟต์แวร์ในการปรับเปลี่ยนสู่ดิจิทัล (The Role of Software Engineer in Digital Transformation) 12
 - 1.5.1 ความสำคัญของวิศวกรซอฟต์แวร์ 12
 - 1.5.2 ความท้าทายของการปรับเปลี่ยนสู่ดิจิทัลในวิศวกรซอฟต์แวร์ 13
 - 1.5.3 แนวทางการเอาชนะความท้าทายของการปรับเปลี่ยนสู่ดิจิทัลในด้านวิศวกรซอฟต์แวร์ 15
- 1.6 คำนิยามและคำจำกัดความที่ใช้ในวิศวกรรมซอฟต์แวร์ 17

CHAPTER

2

การวิเคราะห์เชิงโครงสร้าง STRUCTURED ANALYSIS

23

- 2.1 การวิเคราะห์เชิงโครงสร้าง (Structure Analysis And Design) 24
- 2.2 แผนภาพการไหลของข้อมูล (Data Flow Diagram) 25
 - 2.2.1 แผนภาพบริบท (Context Diagram) 26
 - 2.2.2 แผนภาพการไหลของข้อมูลระดับที่ 0 (DFD Level 0) 27
 - 2.2.3 การแยกองค์ประกอบ (Decomposition) 28
- 2.3 ตัวอย่างการใช้แผนภาพการไหลของข้อมูลกับการปรับรื้อกระบวนการทางธุรกิจ (Using DFD in BPR) 30

CHAPTER

3

การวิเคราะห์และออกแบบเชิงวัตถุ OBJECT-ORIENTED ANALYSIS AND DESIGN

33

- 3.1 กระบวนทัศน์เชิงวัตถุ (Object-Oriented Paradigm) 34
- 3.2 หลักการเชิงวัตถุ (Object-Oriented Concept) 36
- 3.3 นามธรรม (Abstraction) 37
 - 3.3.1 คลาส (Class) และออบเจกต์ (Object) 38

CONTENT

3.3.2 การสื่อสารของออบเจกต์ (Object Communication)	40
3.3.3 ปฏิสัมพันธ์ระหว่างออบเจกต์ (Interacting Objects)	41
3.3.4 การสื่อสารของออบเจกต์ (Object Communication)	42
3.4 การห่อหุ้ม (Encapsulation)	42
3.5 ความเป็นโมดูลาร์ (Modularity)	43
3.6 ลำดับชั้น (Hierarchy)	43
3.6.1 ลำดับชั้นแบบ IS-A	43
3.6.2 ลำดับชั้นแบบ PART-OF	46
3.7 โพลิมอร์ฟิซึม (Polymorphism)	47
3.8 การวิเคราะห์และออกแบบเชิงวัตถุ (Object-Oriented Analysis and Design)	47
3.8.1 บริบทของระบบและรูปแบบการใช้งาน	48
3.8.2 การออกแบบสถาปัตยกรรม	49
3.8.3 การระบุออบเจกต์และคลาส	49
3.8.4 แบบจำลองการออกแบบ	49
3.8.5 กำหนดส่วนต่อประสานของออบเจกต์	50

CHAPTER

4

UML ภาษาการสร้างแบบจำลองแบบครบวงจร

53

4.1 ภาษาการสร้างแบบจำลอง (Modeling Language)	54
--	----

4.2 ความเป็นมาของภาษาการสร้างแบบจำลองแบบครบวงจร (Unified Modeling Language, UML)	55
4.3 แผนภาพยูเอ็มแอล (UML Diagram)	62
4.4 แผนภาพยูสเคส (Use Case Diagrams)	63
4.4.1 องค์ประกอบของยูสเคส (Use Case)	63
4.4.2 ความสัมพันธ์ระหว่างยูสเคส	64
4.4.3 หลักการวิเคราะห์ระบบด้วยยูสเคส	65
4.5 แผนภาพคลาส (Class Diagrams)	68
4.5.1 การวิเคราะห์ความทนทาน (Robustness Analysis)	74
4.5.2 ขั้นตอนในการทำแผนภาพคลาส	76
4.5.3 แนวปฏิบัติที่ดีของการทำแผนภาพคลาส	77
4.6 แผนภาพลำดับ (Sequence Diagrams)	77
4.6.1 ขั้นตอนในการเขียนแผนภาพลำดับ	81
4.6.2 แนวทางปฏิบัติที่ดีของแผนภาพลำดับ	84
4.6.3 กระบวนการสร้างแผนภาพลำดับด้วยการวิเคราะห์ความคงทน	86
4.7 แผนภาพสถานะ (State Diagrams)	87
4.7.1 ขั้นตอนการเขียนแผนภาพสถานะ	89
4.8 แผนภาพกิจกรรม (Activity Diagrams)	91
4.8.1 ขั้นตอนการทำแผนภาพกิจกรรม	92
4.8.2 ตัวอย่างแผนภาพกิจกรรม	94
4.9 แผนภาพการนำไปใช้ (Deployment Diagrams)	95
4.9.1 ขั้นตอนการทำแผนภาพการนำไปใช้	96

4.10 การสร้างรายงานโดยยูเอ็มแอล.....	98
4.11 การพัฒนาโปรแกรมโดยใช้แบบจำลอง.....	100
4.12 การพัฒนาด้วยวิศวกรรมย้อนกลับ (Reverse Engineering)	102

CHAPTER

5

กระบวนการซอฟต์แวร์ SOFTWARE PROCESS

107

5.1 แบบจำลองกระบวนการซอฟต์แวร์ (Software Process Models).....	110
5.1.1 แบบจำลองการพัฒนาระบบแบบน้ำตก (Water Fall Model)	110
5.1.2 แบบจำลองการพัฒนาต้นแบบ (Prototyping Model)	114
5.1.3 แบบจำลองการพัฒนาระบบแบบมีส่วนร่วม (Joint Application Development Model).....	114
5.1.4 แบบจำลองการพัฒนาซอฟต์แวร์อย่างรวดเร็ว (Rapid Application Development Model)	115
5.1.5 การทำงานแบบบอจิล (Agile Methodology)	116
5.1.6 แบบจำลองการพัฒนาโปรแกรมแบบเอ็กซ์ตรีม (Extreme Programming)	124
5.1.7 แบบจำลองการพัฒนาโดยวิธีเชิงวัตถุ (Object Oriented Development)	126
5.1.8 สถาปัตยกรรมเชิงเซอร์วิส (Service Oriented Architecture)	128

5.2 กิจกรรมในกระบวนการซอฟต์แวร์ (Process Activities)	129
5.2.1 การระบุข้อกำหนดของซอฟต์แวร์ (Software Specification)	130
5.2.2 การออกแบบซอฟต์แวร์และการนำไปใช้ (Software Design and Implementation)	132
5.2.3 การตรวจสอบความถูกต้องของซอฟต์แวร์ (Software Validation)	134
5.2.4 วิวัฒนาการของซอฟต์แวร์ (Software Evolution)	135

CHAPTER

6

การจัดการเทคโนโลยีดิจิทัล DIGITAL TECHNOLOGY MANAGEMENT

139

6.1 ปัญหาในการบริหารจัดการองค์กร	141
6.2 สถาปัตยกรรมองค์กร (Enterprise Architecture)	143
6.2.1 สถาปัตยกรรมองค์กรในปัจจุบัน (As is Enterprise Architecture).....	148
6.2.2 สถาปัตยกรรมองค์กรในอนาคต (To be Enterprise Architecture)	149
6.3 สถาปัตยกรรมองค์กรกับวิศวกรรมซอฟต์แวร์ (Enterprise Architecture and Software Engineering).....	154
6.4 การกำกับดูแลการใช้เทคโนโลยีดิจิทัล (Digital Technology Governance)	155

CONTENT

6.5 การจัดการซอฟต์แวร์ (Software Management) 159

6.5.1 การจัดการโครงการด้านซอฟต์แวร์ (Software Project Management) 159

6.5.2 การบริหารโครงการซอฟต์แวร์แบบอไจล์ (Agile Project Management) 165

6.5.3 การจัดการความเสี่ยง (Risk Management) 167

6.5.4 การประมาณการต้นทุนซอฟต์แวร์ (Software Cost Estimation) 172

6.6 การจัดการคุณภาพซอฟต์แวร์ (Software Quality Management)..... 174

6.6.1 การปรับปรุงคุณภาพซอฟต์แวร์ให้มีประสิทธิภาพ 175

6.6.2 CMMI กับกระบวนการพัฒนาแบบอไจล์ 177

6.7 ปัจจัยสู่ความสำเร็จ (Keys Success Factors)..... 178

CHAPTER

7

วิศวกรรมความต้องการ REQUIREMENT ENGINEERING 181

7.1 ความหมายของความต้องการด้านซอฟต์แวร์ (Software Requirement Definition) 183

7.2 กระบวนการวิศวกรรมความต้องการ (Requirement Engineering Process)..... 185

7.3 กระบวนการจัดทำข้อกำหนดความต้องการด้านซอฟต์แวร์ (Software Requirement Specification Process) 189

7.4 การสกัดและวิเคราะห์ความต้องการ (Requirements Elicitation and Analysis).... 190

7.5 การตรวจสอบความต้องการ (Requirements Validation) 222

7.6 การจัดการความต้องการ (Requirements Management) 223

7.7 วิศวกรรมความต้องการแบบอไจล์ (Agile Requirement Engineering)..... 224

CHAPTER

8

การออกแบบระบบ SYSTEM DESIGN 231

8.1 กระบวนการออกแบบระบบ (System Design) 234

8.2 การออกแบบสถาปัตยกรรม (Architectural Design) 235

8.2.1 ลักษณะการออกแบบสถาปัตยกรรมที่ดี ...235

8.2.2 มุมมองการออกแบบสถาปัตยกรรม (Architecture View) 236

8.2.3 รูปแบบสถาปัตยกรรม (Architectural Pattern) 238

8.2.4 สถาปัตยกรรมแอปพลิเคชัน (Application Architecture) 244

8.2.5 การออกแบบสถาปัตยกรรมสำหรับกรณีศึกษา การพัฒนาระบบการออกใบอนุญาตกิจการที่เป็นอันตรายต่อสุขภาพ.....	246
8.3 การออกแบบส่วนต่อประสาน (Interface Design)	250
8.3.1 การออกแบบฟอร์มและรายงาน (Designing Forms and Reports)	250
8.4 การออกแบบส่วนประกอบ (Component Design)	255
8.5 การออกแบบฐานข้อมูล (Database Design) ...	257
8.5.1 ธรรมาภิบาลข้อมูล (Data Governance) ...	258
8.5.2 การบริหารและกระบวนการจัดการข้อมูลหรือวงจรชีวิตของข้อมูล (Data Life Cycle).....	261
8.5.3 การบูรณาการข้อมูล (Data Integration) ...	262
8.6 รูปแบบการออกแบบ (Design Pattern).....	263

CHAPTER

9

การติดตั้งระบบและการบำรุงรักษา SYSTEM IMPLEMENTATION AND MAINTENANCE 269

9.1 การพัฒนาและติดตั้งระบบ (System Implementation)	270
9.1.1 การพัฒนาโปรแกรม (Program Coding) ...	270
9.1.2 การพัฒนด้วยวิศวกรรมย้อนกลับ (Reverse Engineering)	275
9.1.3 วิธีการพัฒนาโปรแกรมแบบ Agile และ DevOps.....	277

9.2 การทดสอบซอฟต์แวร์สำหรับแบบจำลองน้ำตก (Water Fall Model Software Testing).....	281
9.2.1 การทดสอบการพัฒนา (Development Testing).....	282
9.2.2 การทดสอบการเผยแพร่ (Release Testing).....	286
9.2.3 การทดสอบโดยผู้ใช้ (User Testing)	287
9.3 การทดสอบซอฟต์แวร์สำหรับอไจล์ (Agile Software Testing)	290
9.4 การติดตั้งระบบ (System Installation).....	292
9.4.1 การติดตั้งโดยตรง (Direct installation)....	292
9.4.2 การติดตั้งแบบขนาน (Parallel Installation).....	293
9.4.3 การติดตั้งแบบนำร่อง (Single-Location Installation).....	293
9.4.4 การติดตั้งเป็นระยะ (Phase Installation) ..	293
9.5 การจัดทำเอกสาร (Documentation)	294
9.6 การฝึกอบรม (Training)	295
9.7 การบำรุงรักษาซอฟต์แวร์ (Software Maintenance)	295

CHAPTER

10

วิศวกรรมซอฟต์แวร์เชิงเซอร์วิส SERVICE ORIENTED SOFTWARE ENGINEERING 299

10.1 วิศวกรรมซอฟต์แวร์เชิงเซอร์วิส (Service Oriented Software Engineering).....	300
--	------------

CONTENT

10.2	รูปแบบการพัฒนาซอฟต์แวร์เชิงเซอร์วิส.....	304
10.3	เทคโนโลยีเว็บเซอร์วิส (Web Services).....	307
10.3.1	เว็บเซอร์วิสแบบ SOAP (SOAP Web Services).....	308
10.3.2	เว็บเซอร์วิสแบบ REST (REST Web Services).....	315
10.3.3	ข้อแตกต่างระหว่างเว็บเซอร์วิสแบบ SOAP และ REST.....	317
10.4	รูปแบบข้อมูลที่ใช้ในเว็บเซอร์วิส.....	317
10.4.1	ภาษา XML	318
10.4.2	JavaScript Object Notation (JSON)....	318
10.4.3	มาตรฐานการแลกเปลี่ยนข้อมูล.....	319
10.5	สถาปัตยกรรมเชิงเซอร์วิส (Service Oriented Architecture, SOA)	330
10.5.1	การประยุกต์ใช้สถาปัตยกรรมเชิงเซอร์วิส (SOA).....	330
10.5.2	ประโยชน์ในการพัฒนาด้วยเทคโนโลยี SOA	330
10.5.3	ขั้นตอนในการพัฒนาสถาปัตยกรรมเชิงเซอร์วิส (SOA).....	331
10.6	ไมโครเซอร์วิส (Microservices).....	337
10.7	ซอฟต์แวร์คอนเทนเนอร์ (Software Container).....	342

CHAPTER

11

การพัฒนาศักยภาพด้านดิจิทัล DIGITAL HUMAN RESOURCE DEVELOPMENTS 347

11.1	ทักษะด้านดิจิทัล	348
11.2	สมรรถนะของบุคลากรด้านการพัฒนาเทคโนโลยีดิจิทัล	354
11.3	การพัฒนาศักยภาพด้านการพัฒนาเทคโนโลยีดิจิทัล	355
11.3.1	ระดับประธานฝ่ายสารสนเทศ (Chief Information Officer, CIO)	355
11.3.2	ระดับผู้บริหารโครงการ (Project Management Officer, PMO)	356
11.3.3	ระดับผู้พัฒนาระบบ (Developer)	356
11.3.4	ระดับผู้ดูแลระบบ (System Administrator)	358
11.3.5	ระดับผู้ดูแลโครงสร้างพื้นฐาน (Network/Infrastructure)	359

CHAPTER

1

การปรับเปลี่ยนสู่ดิจิทัล DIGITAL TRANSFORMATION

วัตถุประสงค์ (Objectives)

- เพื่ออธิบายรายละเอียดเกี่ยวกับเศรษฐกิจดิจิทัล (Digital Economy)
องค์กรดิจิทัล (Digital Enterprise)
- เพื่ออธิบายแนวคิดในการปรับเปลี่ยนสู่ดิจิทัล (Digital Transformation)
- เพื่อแนะนำหลักการและความสำคัญของวิศวกรรมซอฟต์แวร์ (Software Engineering)
- เพื่ออธิบายการประยุกต์หลักการของวิศวกรรมซอฟต์แวร์กับการพัฒนาโปรแกรมประยุกต์
เพื่อนำไปสู่การปรับเปลี่ยนสู่ดิจิทัล

เนื้อหาประจำบท (Contents)

- 1.1 เศรษฐกิจดิจิทัล (Digital Economy)
- 1.2 ความหมายของเทคโนโลยีดิจิทัล (Digital Technology Definition)
- 1.3 องค์กรดิจิทัล (Digital Enterprises)
- 1.4 การปรับเปลี่ยนสู่ดิจิทัล (Digital Transformation)
- 1.5 บทบาทของวิศวกรรมซอฟต์แวร์กับการปรับเปลี่ยนสู่ดิจิทัล
- 1.6 คำนิยามและคำจำกัดความที่ใช้ในวิศวกรรมซอฟต์แวร์

1.1 เศรษฐกิจดิจิทัล (Digital Economy)

ผู้ที่บัญญัติศัพท์นี้เป็นคนแรกคือ Don Tapscott ในหนังสือเรื่อง The Digital Economy [5] ในปี ค.ศ. 1995 และเขียนเพิ่มเติมในปี ค.ศ. 2015 ศึกษาประวัติเพิ่มเติมได้ที่เว็บไซต์ <https://dontapscott.com>

เศรษฐกิจดิจิทัล (Digital Economy, DE) คือ เศรษฐกิจและสังคมที่ใช้เทคโนโลยีดิจิทัลเป็นกลไกสำคัญในการขับเคลื่อนการปฏิรูปกระบวนการผลิต การดำเนินธุรกิจ การค้า การบริการ การศึกษา การสาธารณสุข การบริหารราชการแผ่นดิน รวมทั้งกิจกรรมทางเศรษฐกิจและสังคมอื่นๆ ที่ส่งผลต่อการพัฒนาทางเศรษฐกิจ การพัฒนาคุณภาพชีวิตของคนในสังคมและการจ้างงานที่เพิ่มขึ้น

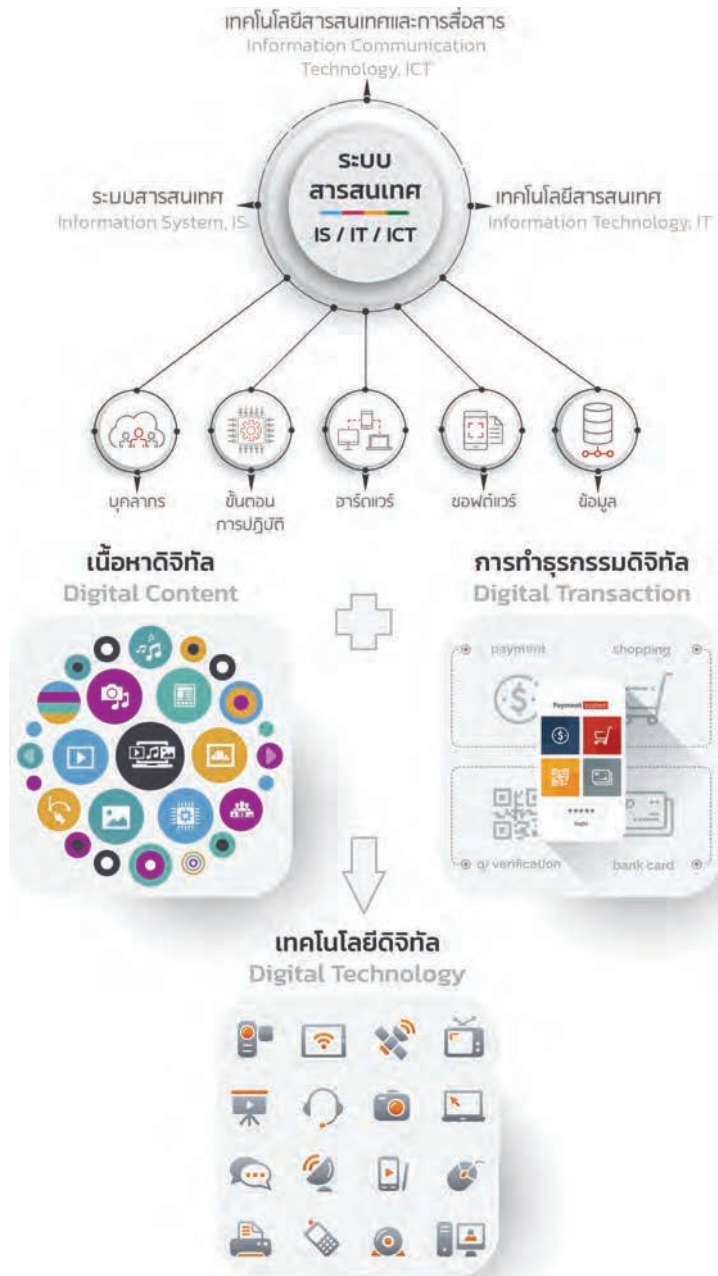
คุณลักษณะที่สำคัญของเศรษฐกิจดิจิทัล มี 12 ประการ คือ

- 1. องค์ความรู้ (Knowledge)** เศรษฐกิจยุคใหม่คือเศรษฐกิจแห่งองค์ความรู้ โดยที่ปัญญาประดิษฐ์ (Artificial Intelligence, AI) จะเข้ามามีบทบาททำให้อุตสาหกรรมการผลิตและบริการเปลี่ยนรูปแบบไป ทำให้กระทบต่อการจ้างงาน และการปรับตัวของธุรกิจ ดังนั้น การทำงานทุกอย่างต้องนำไปสู่องค์ความรู้และทรัพย์สินทางปัญญา
- 2. ปรับเปลี่ยนสู่ดิจิทัล (Digital Transformation)** กิจกรรมทุกอย่างทั้งในด้านธุรกิจ การทำงาน การศึกษา จะต้องถูกปรับเปลี่ยนเป็นดิจิทัล เพื่อตอบสนองความต้องการของผู้ใช้โดยไม่จำกัดสถานที่และเวลา ซึ่งจะเห็นได้อย่างชัดเจน ในช่วงที่มีการระบาดของเชื้อไวรัสโคโรนา (COVID-19) องค์กรมีการใช้เทคโนโลยีดิจิทัล เช่น การประชุมทางไกล (Tele-conference) หรือการทำงานที่บ้าน (Work at Home) เพิ่มขึ้นอย่างมาก ซึ่งการเปลี่ยนแปลงสู่การทำงานในโลกดิจิทัลมีส่วนช่วยให้องค์กรสามารถอยู่รอดได้
- 3. ความเสมือนจริง (Virtual)** เมื่อข้อมูลถูกเปลี่ยนรูปไปเป็นดิจิทัล สิ่งที่เคยจับต้องได้ก็จะกลายเป็นเสมือนจริง ซึ่งจะกระทบถึงกิจกรรมหลายอย่างที่เกี่ยวข้องกับการทำธุรกิจและการดำเนินชีวิต เช่น ตลาดเสมือน (Virtual Market) สำนักงานเสมือน (Virtual Office) เป็นต้น
- 4. ขับเคลื่อนด้วยหน่วยงานย่อย** เป็นการขับเคลื่อนเศรษฐกิจด้วยหน่วยย่อยที่มีอิสระ สามารถเข้ามาร่วมทำกิจกรรมตามที่ตัวเองถนัด
- 5. เครือข่ายบูรณาการ** เทคโนโลยีดิจิทัลทำให้ลดช่องว่างทั้งในองค์กรและระหว่างองค์กร ทำให้เกิดการบูรณาการการทำงานหน่วยงานย่อยมารวมกัน เพื่อขับเคลื่อนเศรษฐกิจในภาพรวมได้อย่างมีประสิทธิภาพ
- 6. ปราศจากคนกลาง** ในเศรษฐกิจดิจิทัลคนกลางที่ทำงานอยู่ระหว่างผู้ผลิตและผู้บริโภคจะหายไป เช่น ธุรกิจแบบอีคอมเมิร์ซ ทำให้ผู้ซื้อและผู้ขายสามารถซื้อ-ขายกันได้โดยตรง เป็นต้น
- 7. หลอมรวมเป็นหนึ่ง** ภาคส่วนต่างๆ จะประสบความสำเร็จได้โดยการสนับสนุนจากภาคส่วนอื่นๆ ที่อยู่รอบตัว โดยใช้เทคโนโลยีดิจิทัล

8. **นวัตกรรม (Innovation)** สินค้า บริการ กระบวนการ ระบบงาน การตลาด และประชาชน ต้องมีนวัตกรรม เพื่อตอบสนองความต้องการใหม่ๆ ของลูกค้า เช่น การตลาดดิจิทัล เป็นต้น
9. **ผู้บริโภคร่วมผลิต** การผลิตสินค้าจะเปลี่ยนเป็นการผลิตตามความต้องการของลูกค้าแทนที่การผลิตจำนวนมากๆ เช่น การผลิตสินค้าตามความต้องการของลูกค้าด้วยเครื่องพิมพ์ 3 มิติ เป็นต้น
10. **องค์กรตอบสนองทันกาล (Real Time Enterprise)** องค์กรยุคใหม่ต้องปรับปรุงตนเองให้มีความทันสมัยอยู่เสมอ ระบบแลกเปลี่ยนข้อมูลทางอิเล็กทรอนิกส์ (Electronic Data Interchange, EDI) จะมีส่วนอย่างมากในการเชื่อมโยงข้อมูลและสารสนเทศ เพื่อให้องค์กรสามารถรับรู้และปรับปรุงข้อมูลให้ทันสมัยตลอดเวลา
11. **โลกาภิวัตน์ (Globalization)** ด้วยระบบเครือข่ายอินเทอร์เน็ตทำให้โลกแคบลง การทำธุรกิจจึงไร้ขอบเขต เช่น การใช้เครือข่ายสังคม (Social Network) เป็นต้น
12. **ความขัดแย้งไม่ลงรอยกัน** การเติบโตอย่างก้าวกระโดดของเครือข่ายสังคม (Social Network) ได้สร้างความบาดหมางขึ้นในสังคม ทั้งในเรื่องความเป็นส่วนตัว คุณภาพการทำงาน และการใช้ชีวิตส่วนตัว เช่น การรังแก กลั่นแกล้งผู้อื่น (Bully) ในอินเทอร์เน็ต เป็นต้น

1.2 ความหมายของเทคโนโลยีดิจิทัล (Digital Technology Definition)

ก่อนที่จะเข้าใจความหมายของเทคโนโลยีดิจิทัล (Digital Technology) ควรจะเข้าใจความหมายของคำว่า “ข้อมูล” (Data) หมายถึง ข้อเท็จจริงต่างๆ ที่เกิดขึ้นแล้วในอดีตถึงปัจจุบัน และ “สารสนเทศ” (Information) หมายถึง ข้อมูลที่ผ่านการประมวลผลด้วยวิธีการต่างๆ เพื่อให้ได้ผลลัพธ์ตรงตามความต้องการของผู้ใช้สารสนเทศ ทั้งข้อมูลและสารสนเทศจะได้รับการจัดการใน “ระบบสารสนเทศ” (Information System, IS) แม้จะไม่มีระบบคอมพิวเตอร์ก็ตาม เช่น การบันทึกใส่สมุดหรือกระดาษ เป็นต้น ต่อมาเมื่อมีการพัฒนาระบบคอมพิวเตอร์ที่ทันสมัย ก็ได้มีการนำมาใช้ในการจัดการสารสนเทศที่เรียกว่า “เทคโนโลยีสารสนเทศ” (Information Technology, IT) ซึ่งประกอบไปด้วย ส่วนนำเข้า ส่วนประมวลผล และส่วนแสดงผล ซึ่งข้อมูลจะเป็นวัตถุดิบของระบบในส่วนนำเข้าเพื่อประมวลผลด้วยวิธีต่างๆ และได้สารสนเทศเป็นผลลัพธ์ออกมา ซึ่งเทคโนโลยีสารสนเทศนี้ช่วยให้การประมวลผลทำได้รวดเร็วขึ้น และสารสนเทศที่ได้ก็มีคุณภาพน่าเชื่อถือมากขึ้น ต่อมาด้วยเทคโนโลยีการสื่อสารที่พัฒนาไปอย่างรวดเร็วจึงมีการรับ-ส่งข้อมูล และสารสนเทศในปริมาณที่มากขึ้นจึงเปลี่ยนเป็นคำว่า “เทคโนโลยีสารสนเทศและการสื่อสาร” (Information Communication Technology, ICT) และเมื่อการใช้อินเทอร์เน็ตและเครือข่ายสังคม (Social Network) มากขึ้น จึงเกิด “เนื้อหาดิจิทัล” (Digital Content) และ “การทำธุรกรรมดิจิทัล” (Digital Transaction) ผ่านระบบเครือข่ายซึ่งเรียกว่า “เทคโนโลยีดิจิทัล” (Digital Technology) ดังแสดงในรูปที่ 1.1



รูปที่ 1.1 องค์ประกอบของเทคโนโลยีดิจิทัล

1.3 องค์กรดิจิทัล (Digital Enterprises)

องค์กรดิจิทัล (Digital Enterprises) หมายถึง องค์กรที่ขับเคลื่อนธุรกิจด้วยเทคโนโลยีดิจิทัล ประกอบด้วย 4 องค์ประกอบ [8] คือ

1. องค์กรมีรูปแบบธุรกิจแบบดิจิทัล (Digital Business Models)

องค์กรดิจิทัลจะต้องมีการปรับเปลี่ยนรูปแบบธุรกิจ (Business Model) ให้ใช้เทคโนโลยีดิจิทัลและนวัตกรรมทั้งในด้านการผลิตสินค้าและการบริการ เช่น การขยายช่องทางการขาย การผลิตสินค้า การตลาดดิจิทัล เป็นต้น ดังนั้น องค์กรจึงต้องมีการจัดทำแผนพัฒนาดิจิทัล เพื่อกำหนดวิสัยทัศน์ พันธกิจ และยุทธศาสตร์ให้สอดคล้องกับเศรษฐกิจดิจิทัล รวมถึงแผนพัฒนาดิจิทัลเพื่อเศรษฐกิจและสังคมของประเทศไทย

2. องค์กรมีรูปแบบการดำเนินงานแบบดิจิทัล (Digital Operating Models)

รูปแบบการดำเนินงานขององค์กรดิจิทัลต้องมีการจัดระเบียบบุคลากร กระบวนการ และเทคโนโลยี เพื่อให้บรรลุวัตถุประสงค์เชิงกลยุทธ์ รูปแบบการดำเนินงานแบบดิจิทัลต้องให้ความสำคัญกับลูกค้า หรือผู้ใช้บริการเป็นหลัก และเชื่อมต่อช่องว่างระหว่างผู้บริหารระดับสูง ฝ่ายเทคโนโลยีสารสนเทศ และฝ่ายปฏิบัติการ เพื่อตอบสนองความต้องการและความคาดหวังของลูกค้า หรือผู้ใช้บริการ โดยที่กระบวนการสำคัญ โครงสร้างองค์กร และโครงสร้างพื้นฐาน ได้รับการออกแบบเพื่อรองรับการเดินทางของลูกค้า (Customer Journey)

3. องค์กรมีการติดตามประเมินผลแบบดิจิทัล (Digital Traction Metrics)

คือ การกำหนดวิธีการตรวจสอบกลยุทธ์ดิจิทัลขององค์กร เช่น รูปแบบธุรกิจดิจิทัล การดำเนินงานดิจิทัล ความสามารถและทักษะด้านดิจิทัลของบุคลากร

4. บุคลากรในองค์กรมีความสามารถและทักษะดิจิทัล (Digital Talent and Skills)

องค์ประกอบที่สำคัญขององค์กรดิจิทัล คือ บุคลากรที่มีความรู้ความสามารถในการใช้เทคโนโลยีดิจิทัลที่มีอยู่ปรับตัวให้เข้ากับวิธีการที่กำลังพัฒนาและแนวทางใหม่ๆ เช่น หุ่นยนต์ปัญญาประดิษฐ์ วิทยาศาสตร์ข้อมูล ความจริงเสมือน และรูปแบบธุรกิจดิจิทัลใหม่ๆ บนอินเทอร์เน็ต ซึ่งจะเข้ามาแทนที่วิธีการทำงานแบบเดิม (Digitalization)

1.4 การปรับเปลี่ยนสู่ดิจิทัล (Digital Transformation)

การปรับเปลี่ยนสู่ดิจิทัล (Digital Transformation) คือ การปรับเปลี่ยนองค์กรเพื่อให้เข้ากับเศรษฐกิจดิจิทัล (Digital Economy) ด้วยเทคโนโลยีดิจิทัล (Digital Technology) ที่เป็นการปรับเปลี่ยนองค์กรให้ใช้ประโยชน์จากเทคโนโลยีดิจิทัล (Digitalization) ตั้งแต่การกำหนดเป้าหมาย วางแผน พัฒนาผลิตภัณฑ์หรือบริการ ตลอดจนช่องทางการส่งมอบสินค้าและบริการให้กับผู้บริโภค เพื่อเปลี่ยนองค์กรให้เป็นองค์กรดิจิทัลที่มีกระบวนการความคิดแบบดิจิทัล (Digital Mindset) และพัฒนาองค์กรให้เติบโตด้วยเทคโนโลยีดิจิทัล



รูปที่ 1.2 ขั้นตอนในการปรับเปลี่ยนสู่ดิจิทัล [2]

1.4.1 ขั้นตอนการปรับเปลี่ยนสู่ดิจิทัล (Digital Transformation Roadmap)

แผนงานสู่การปรับเปลี่ยนสู่ดิจิทัล Earley Information Science, Inc. สรุปกระบวนการไว้ 4 ขั้นตอน [1] ดังแสดงในรูปที่ 1.3 คือ

STEP 1 Current State Assessment

สำรวจและประเมินสถานภาพปัจจุบันขององค์กรอย่างรอบด้าน

STEP 2 Future Vision

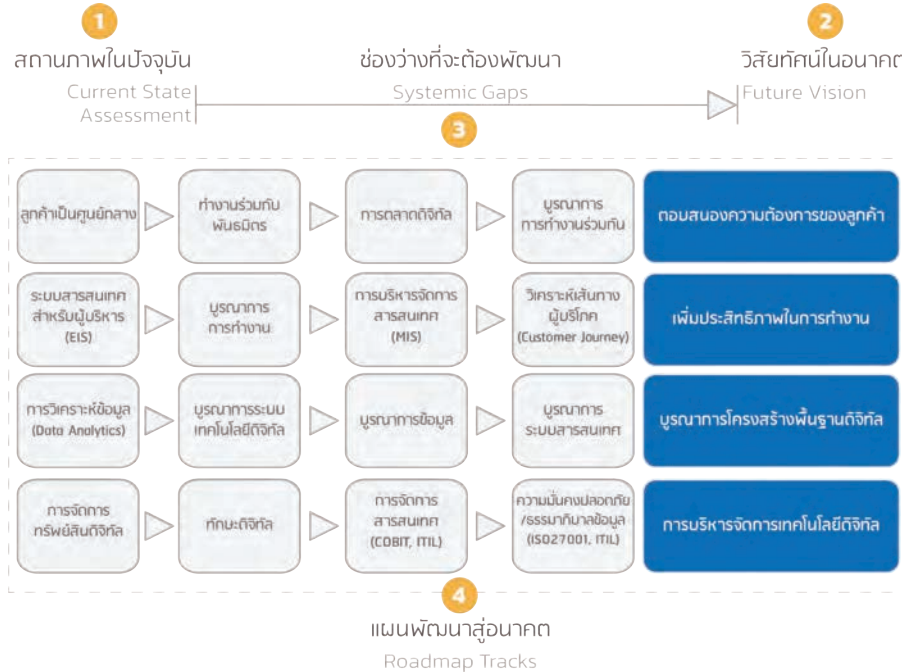
กำหนดวิสัยทัศน์ในอนาคตสำหรับการดำเนินงานแบบดิจิทัลอย่างชัดเจน

STEP 3 Systemic Gaps

วิเคราะห์ช่องว่างระหว่างสถานภาพในปัจจุบันกับวิสัยทัศน์ในอนาคต

STEP 4 Roadmap Tracks

สร้างแนวทางหรือกลยุทธ์ตามปัจจัย 4 คือ 1) บุคลากร (People) 2) กระบวนการทำงาน (Process) 3) เทคโนโลยีดิจิทัล (Digital Technology) และ 4) การบริหารจัดการเทคโนโลยีดิจิทัล (Digital Technology Management) และธรรมาภิบาล (Governance)



รูปที่ 1.3 ช่องว่าง (Gap) ที่ต้องพัฒนาเพื่อการปรับเปลี่ยนสู่ดิจิทัล

ผู้บริหารในองค์กรเข้าใจผิดคิดว่า การปรับเปลี่ยนสู่ดิจิทัล (Digital Transformation) คือ เรื่องของเทคโนโลยี เป็นเรื่องของแผนกคอมพิวเตอร์ หรือเทคโนโลยีสารสนเทศ และคิดว่าการลงทุนด้านเทคโนโลยีดิจิทัลทำให้เกิดการพัฒนาไปสู่การเป็น องค์กรดิจิทัล (Digital Enterprise) ได้ ทั้งๆ ที่โดยแท้จริงแล้ว การปรับเปลี่ยนสู่ดิจิทัลคือเรื่องของ กลยุทธ์องค์กร สิ่งที่ต้องพัฒนาเพื่อให้บรรลุเป้าหมายที่ต้องการคือ “ดิจิทัลไเซชัน” (Digitization) และ “ดิจิทัลไลเซชัน” (Digitalization) การพัฒนาระบบด้วยหลักการทางวิศวกรรมซอฟต์แวร์เพียงอย่างเดียวจะเป็นเพียงการแปลงข้อมูลต่างๆ ให้อยู่ในรูปดิจิทัล อาจเป็นเพียงขั้นตอนของการทำดิจิทัลไเซชัน ดังนั้น ถ้าจะทำดิจิทัลไลเซชันซึ่งคือ การนำเทคโนโลยีดิจิทัลเข้ามา แล้วทำให้กระบวนการทำงานเปลี่ยนไป มีการทำงานที่มีประสิทธิภาพขึ้น ข้อสำคัญคือ “การเปลี่ยนประสบการณ์ของลูกค้า” (Customer Experience) ผ่าน “การเดินทางของลูกค้า” (Customer Journey) ซึ่งการปรับเปลี่ยนสู่ดิจิทัลก็คือผลลัพธ์ที่ได้จากการทำดิจิทัลไลเซชันนั่นเอง

ตัวอย่างเช่น การให้บริการของหน่วยงานส่วนใหญ่มุ่งเน้นการพัฒนาบริการทางอิเล็กทรอนิกส์ (E-Services) โดยการวิเคราะห์ และพัฒนาจากกระบวนการทำงานแบบเดิมๆ บริบทเดิมๆ ซึ่งทำให้ได้ซอฟต์แวร์ที่ทำงานเหมือนเดิม เพียงเปลี่ยนจากการกรอกในแบบฟอร์มที่เป็นกระดาษมาเป็นกรอกผ่านเว็บเท่านั้น ส่วนการทำงานอื่นยังเหมือนเดิม เช่น ต้องเดินทางไปที่หน่วยงานเพื่อชำระเงิน หรือต้องไปธนาคารเพื่อโอนเงิน เป็นต้น หน่วยงานยังคงต้องการเอกสารมากมาย เช่น สำเนาบัตรประชาชน ทะเบียนบ้าน โดยให้ประชาชนอัปโหลดเอกสารเหล่านั้นมาทางอินเทอร์เน็ต เป็นต้น



ลักษณะที่กล่าวมาเป็นเพียงการทำดิจิทัลเซชันเท่านั้น ยังไม่อาจจะเรียกได้ว่าเป็นดิจิทัลไลเซชัน เพราะยังคงกระบวนการทำงานเหมือนเดิม ใช้ทรัพยากรเท่าเดิม และไม่ได้ตอบสนองความต้องการของผู้มีส่วนได้ส่วนเสีย เพราะการวิเคราะห์ความต้องการนั้นเน้นที่บุคลากรของหน่วยงานเป็นหลัก ไม่ได้ไปวิเคราะห์ความต้องการของผู้มีส่วนได้ส่วนเสีย ทั้งที่ในรูปแบบเดิมๆ อาจมีล่าช้าและซับซ้อน ซึ่งหน่วยงานสามารถใช้เทคโนโลยีดิจิทัลทำให้ประสบการณ์ของลูกค้าเปลี่ยนไปในทางที่ดีขึ้นได้

1.4.2 กรอบการปรับเปลี่ยนสู่ดิจิทัล (Digital Transformation Framework)

กรอบการปรับเปลี่ยนสู่ดิจิทัล [1] มีองค์ประกอบดังรูปที่ 1.4



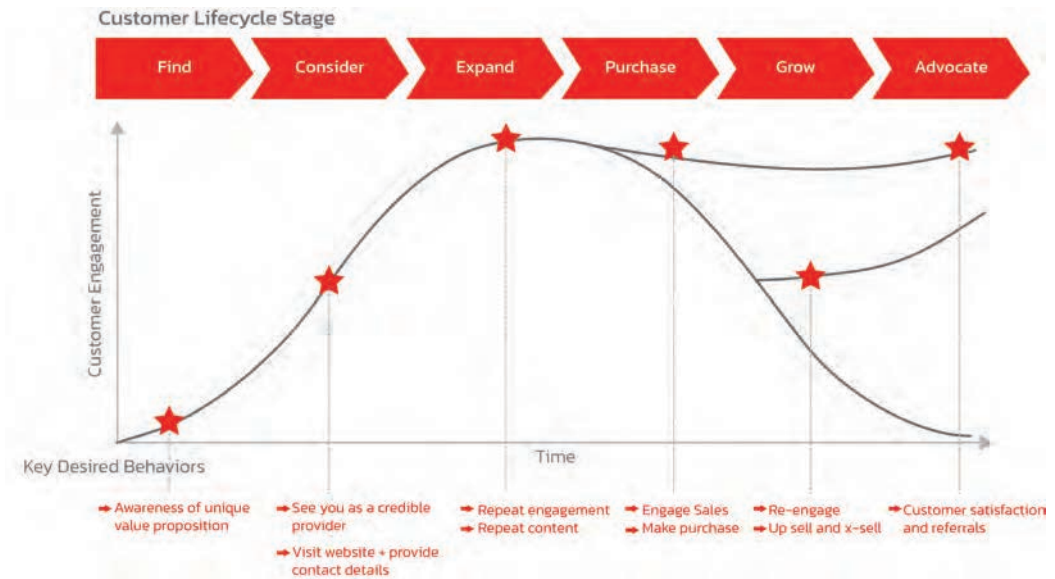
รูปที่ 1.4 กรอบการปรับเปลี่ยนสู่ดิจิทัล [1]

1. กลยุทธ์ทางธุรกิจที่ขับเคลื่อนด้วยดิจิทัล (Business Strategy Driven by Digital)

โดยจะต้องมองแนวโน้มด้านดิจิทัลที่จะมีผลต่อธุรกิจหรือการทำงาน เช่น การใช้สารสนเทศในการเชื่อมโยงกับผู้มีส่วนได้ส่วนเสีย การประยุกต์ใช้เทคโนโลยีดิจิทัลเปลี่ยนแปลงองค์กรด้านต่างๆ หรือการพัฒนาผลิตภัณฑ์หรือบริการที่เป็นรูปแบบดิจิทัล (กลยุทธ์ด้านดิจิทัลต้องสอดคล้องกับกลยุทธ์ขององค์กร)

2. การนำเทคโนโลยีดิจิทัลเข้ามาใช้ในองค์กร (Digitize the Core)

เพื่อให้ธุรกิจหลักมีประสิทธิภาพมากขึ้น เช่น การวางระบบบริหารทรัพยากรขององค์กร (Enterprise Resource Planning, ERP) หรือระบบลูกค้าสัมพันธ์ (Customer Relationship Management, CRM) เป็นต้น ขณะเดียวกันก็ต้องนำระบบดิจิทัลเข้าไปวิเคราะห์กระบวนการตัดสินใจของผู้บริโภคในการเลือกซื้อสินค้า หรือบริการ (Customer Journey) โดยพัฒนากลยุทธ์และช่องทางในการสื่อสารกับลูกค้าเพื่อเปลี่ยนประสบการณ์ของลูกค้า จากรูปที่ 1.5 แสดงสเตจของวงจรชีวิตลูกค้า (Customer Lifecycle Stage) ซึ่งเป็นพฤติกรรม การตัดสินใจซื้อสินค้าของลูกค้าโดยส่วนใหญ่



รูปที่ 1.5 การเดินทางของลูกค้า (Customer Journey) ในการตัดสินใจเลือกซื้อสินค้าหรือบริการ [2]

3. แสวงหาการเติบโตทางดิจิทัล (Seeking Digital Growth)

องค์กรจะต้องเปลี่ยนแปลงตัวเองด้วยการหาแนวทางการเติบโตของธุรกิจในรูปแบบใหม่ ที่ผ่านสินค้าหรือบริการที่มีนวัตกรรมด้านเทคโนโลยีดิจิทัลมากขึ้น ซึ่งบางครั้งก็อาจต้องกระทบต่อการลงทุนธุรกิจในรูปแบบเดิมๆ ที่อาจเปลี่ยนแปลงไป ซึ่งการลงทุนในด้านนี้ก็ต้องสร้างหน่วยธุรกิจใหม่ๆ หรือร่วมกับพันธมิตรทางธุรกิจ

4. ปัจจัยสำคัญที่จะทำให้การปรับเปลี่ยนสู่ดิจิทัลประสบความสำเร็จ ประกอบด้วย

4.1 บุคลากรที่มีความรู้และทักษะด้านเทคโนโลยีดิจิทัล (People/Organization) องค์กรต้องมีความคล่องตัว (Agility) และต้องสร้างวัฒนธรรมดิจิทัล (Digital Culture) ในองค์กร ทั้งนี้ ทรัพยากรมนุษย์ถือเป็นปัจจัยที่สำคัญที่สุด โดยให้องค์กรประเมินสถานการณ์ปัจจุบันในด้านนี้ ได้แก่

4.1.1 ด้านวัฒนธรรมลูกค้าเป็นศูนย์กลางโดยพิจารณา ดังนี้

- มีการใช้ข้อมูลเชิงลึกของลูกค้า (Data Insight) ให้เกิดประโยชน์หรือไม่
- มีการใช้เครื่องมือในการวิเคราะห์พฤติกรรมลูกค้าในทุกช่องทาง (Omni-Channels) เพื่อการตลาดแบบดิจิทัล (Digital Marketing) แดไหน
- เชื่อมโยงข้อมูลลูกค้าให้เป็นแบบองค์รวม (Unified Customer Data) หรือไม่
- มีเครื่องมือที่เหมาะสมในการวัดความพึงพอใจของลูกค้าในทุกช่องทางหรือยัง
- มีการจัดกลุ่มลูกค้า (Segmentation) ตามความสนใจหรือไม่



4.1.2 ด้านความร่วมมือกับพันธมิตร (Partner Enable)

- มีการใช้เทคโนโลยีในการบริหารห่วงโซ่อุปทาน (Supply Chain) หรือยัง
- มีการบริหารประสบการณ์ของลูกค้า (Customer Experience) ผ่านลูกค้าแคโทน และมีการแลกเปลี่ยนข้อมูลกับพันธมิตรหรือไม่

4.1.3 ด้านการมีส่วนร่วมของผู้ซื้อรายบุคคล (Personalized Buyer Engagement)

- มีการจัดทำข้อความผ่านทางเครื่องมือทางการตลาดและสื่อสารให้กับลูกค้ากลุ่มต่างๆ หรือยัง
- มีการจัดเรียงหรือให้ความสำคัญในการให้บริการลูกค้าประเภทต่างๆ แคโทน
- มีการคัดกรองลูกค้าที่มีศักยภาพในการใช้เครื่องมือและข้อมูลทั้งจากภายในองค์กรและพันธมิตรหรือไม่

4.1.4 ด้านวัฒนธรรมองค์กรในการทำงานข้ามสายงาน

- พนักงานยึดติดกับการทำงานเฉพาะในแผนกของตัวเองแคโทน
- ตัวชี้วัดแต่ละแผนกยังแยกกันเป็นเอกเทศ หรือสอดคล้องกับเป้าหมายขององค์กรในอนาคต

4.2 การรวบรวมข้อมูลจากแหล่งต่างๆ และนำข้อมูลมาใช้ในการวิเคราะห์ (Data & Analytics) เพื่อสร้างความได้เปรียบเชิงธุรกิจและมีเป้าหมายการใช้ข้อมูลที่ชัดเจน รวมทั้งเพิ่มประสิทธิภาพในการทำงานโดยพิจารณาปัจจัยต่างๆ ดังนี้

4.2.1 มีการใช้เครื่องมือประเภทธุรกิจอัจฉริยะ (Business Intelligence, BI) การวิเคราะห์ข้อมูลเพื่อช่วยในการตัดสินใจเชิงธุรกิจหรือทำนายแนวโน้มผลลัพธ์การทำงานให้แต่ละฝ่ายทั้งด้านพฤติกรรมลูกค้าและรายได้หรือไม่

4.2.2 มีการใช้ปัญญาประดิษฐ์ (Artificial Intelligence, AI) มาช่วยในการคาดการณ์แบบจำลองข้อมูล (Predictive Data Model) หรือไม่

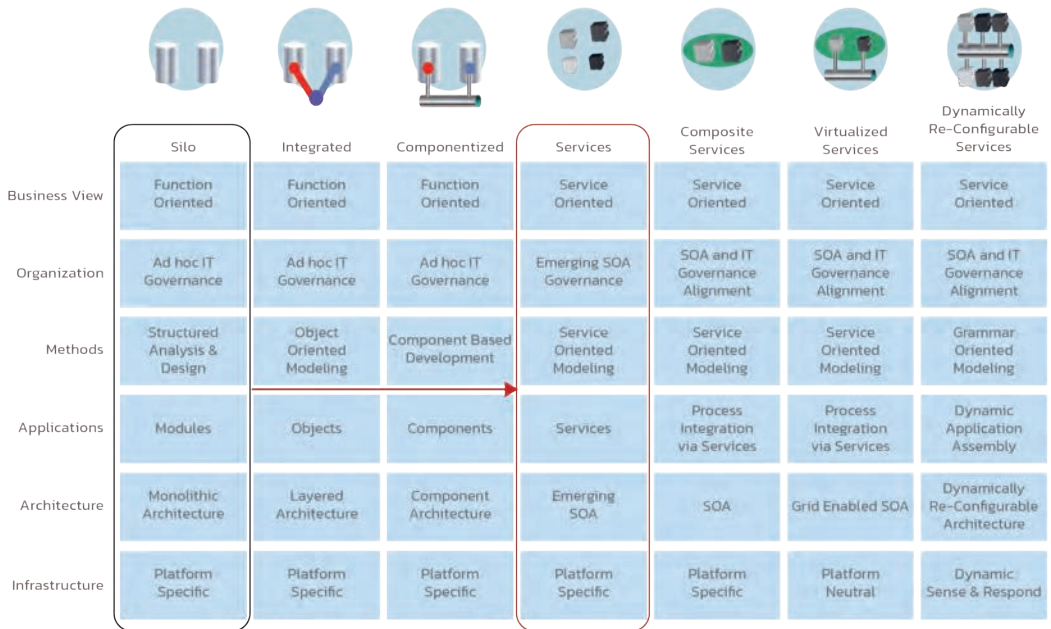
4.2.3 มีการบูรณาการทำงานทั้งองค์กรและมีการกำหนดข้อตกลงการให้บริการ (Service Level Agreement, SLA) ให้พนักงานที่ต้องบริการลูกค้าหรือยัง

4.2.4 มีการทำธรรมาภิบาลดิจิทัล (Digital Governance) เพื่อวางมาตรฐานการทำงานในยุคดิจิทัล การทำความสะอาดข้อมูล (Cleansing Data) และธรรมาภิบาลข้อมูล (Data Governance) ทั้งองค์กรหรือไม่

4.3 เทคโนโลยี (Technology) ในองค์กรที่จะสามารถทำงานได้อย่างมีประสิทธิภาพ ตลอดจนอาจต้องใช้เทคโนโลยีใหม่ๆ เช่น เทคโนโลยีข้อมูลขนาดใหญ่ (Big Data) ที่สอดคล้องกับความจำเป็นที่จะต้องเปลี่ยนแปลงองค์กรให้สามารถสร้างประสบการณ์ใหม่ให้กับลูกค้าได้ดีขึ้น

4.3.1 มีการบูรณาการเชื่อมโยงระบบเทคโนโลยีสารสนเทศเข้าด้วยกันแคโทน

4.3.2 มีการบูรณาการข้อมูลที่จัดเก็บอยู่ต่างระบบ (Silos) รวมถึงมีการรวบรวมข้อมูลเพื่อใช้ประโยชน์ในการทำงานร่วมกันแคโทน



รูปที่ 1.6 การปรับเปลี่ยนสู่ดิจิทัลในบริบทเทคโนโลยีสารสนเทศจากแบบโซลูแบบเซอร์วิส

(ที่มา : www.ibm.com/software/solution/soa)

4.3.3 มีโครงสร้างพื้นฐานด้านเทคโนโลยีดิจิทัลที่รองรับการทำงานตามบริบทของระบบงานแต่ละ

4.4 ระบบนิเวศ (Ecosystem) องค์กรที่ประสบความสำเร็จในการปรับเปลี่ยนสู่ดิจิทัลส่วนมากจะไม่ได้ทำงานตามลำพัง แต่อาจอาศัยพันธมิตรในระบบนิเวศดิจิทัล และมีการสร้างแพลตฟอร์มทางธุรกิจที่หลากหลาย ฝ่ายมีส่วนร่วมและได้รับประโยชน์ทุกฝ่าย

4.4.1 การจัดการทรัพย์สินดิจิทัล (Digital Asset Management) มีการใช้เครื่องมือในการบริหารทรัพย์สินทางปัญญา เช่น ข้อความ ไฟล์รูปภาพ เนื้อหาดิจิทัล เป็นต้น หรือไม่

4.4.2 การจัดการสารสนเทศผลิตภัณฑ์ (Product Information Management) มีการจัดการข้อมูลผลิตภัณฑ์ไว้เป็นระบบ เพื่อนำเสนอลูกค้าได้อย่างเหมาะสมหรือไม่

4.4.3 การเพิ่มประสิทธิภาพการค้นหาข้อมูลในองค์กร มีการใช้เครื่องมือค้นหาที่ดีเพื่อให้สามารถเข้าถึงสารสนเทศต่างๆ ภายในหน่วยงานได้อย่างรวดเร็ว และระบบจัดการความรู้ (Knowledge Management) ของทั้งองค์กรหรือไม่

ทั้งนี้ แนวทางการปรับเปลี่ยนสู่ดิจิทัล อาจเริ่มต้นในลักษณะการถ่ายทอดจากผู้บริหารสูงสุด (Top-Down) หรืออาจเริ่มจากหน่วยธุรกิจใดธุรกิจหนึ่ง (Bottom-Up) หรือบางครั้งก็อาจเริ่มต้นด้วยการวิเคราะห์การเดินทางของลูกค้า (Customer Journey) แล้วมาพิจารณาว่าเทคโนโลยีดิจิทัลจะช่วยให้เกิดการเปลี่ยนแปลงได้อย่างไร



1.5 บทบาทของวิศวกรซอฟต์แวร์ในการปรับเปลี่ยนสู่ดิจิทัล (The Role of Software Engineer in Digital Transformation)

การเปลี่ยนแปลงของเทคโนโลยีดิจิทัลก่อให้เกิดการปรับเปลี่ยนวิถีชีวิตในทุกๆ ด้าน เริ่มจากผลกระทบทางธุรกิจ ซึ่งปรับเปลี่ยนจากโมเดลธุรกิจ (Business Model) แบบเดิมๆ ไปสู่การทำธุรกิจที่ใช้เทคโนโลยีดิจิทัล เช่น ธุรกิจอีคอมเมิร์ซ เป็นต้น บริษัทต่างๆ จำเป็นต้องตอบสนองต่อการปรับเปลี่ยนสู่ระบบดิจิทัล (Digital Transformation) ที่เพิ่มขึ้น และต้องปรับตัวอยู่ตลอดเวลา แนวคิดในการสร้างโมเดลธุรกิจใหม่ หรือปรับเปลี่ยนโมเดลธุรกิจที่มีอยู่ต้องอิงจากการปรับเปลี่ยนสู่ดิจิทัลที่เพิ่มขึ้นในสังคม เช่น การใช้อุปกรณ์เคลื่อนที่ เครือข่ายสังคม และอินเทอร์เน็ตของสรรพสิ่ง (Internet of Things, IoT) เป็นต้น นอกจากนี้เทคโนโลยีดิจิทัลยังส่งผลกระทบต่อชีวิตประจำวัน เช่น การใช้แอปพลิเคชันเพื่อทำธุรกรรมต่างๆ การเดินทาง หรือแม้กระทั่งการรักษาสุขภาพ

1.5.1 ความสำคัญของวิศวกรซอฟต์แวร์

ในปัจจุบันทุกประเทศล้วนขึ้นกับระบบที่มีความซับซ้อน ทั้งระบบโครงสร้างพื้นฐานและระบบสาธารณูปโภคต่างๆ รวมถึงการทำธุรกิจและการดำรงชีวิตประจำวันส่วนใหญ่จะใช้ระบบคอมพิวเตอร์เป็นตัวควบคุมทั้งสิ้น ซึ่งล้วนใช้ซอฟต์แวร์ในการควบคุมการทำงานเป็นส่วนสำคัญ ดังนั้น เศรษฐกิจของประเทศที่เจริญแล้วจึงขึ้นอยู่กับซอฟต์แวร์เป็นหลัก

วิศวกรซอฟต์แวร์ (Software Engineering) จะคำนึงถึงทฤษฎี วิธีการ และเครื่องมือ เพื่อพัฒนาซอฟต์แวร์แบบมืออาชีพผ่านกิจกรรมการสร้างแบบจำลอง (Model) โดยวิศวกรซอฟต์แวร์จะจัดการกับความซับซ้อนด้วยการสร้างแบบจำลองที่มุ่งเน้นในรายละเอียดที่เกี่ยวข้องเท่านั้น และไม่สนใจทุกสิ่งในระหว่างการพัฒนา โดยวิศวกรซอฟต์แวร์จะสร้างแบบจำลองที่แตกต่างกันตามความต้องการของระบบและผู้ใช้ ภายใต้ข้อจำกัดด้านบุคลากร เวลา และงบประมาณ

วิธีการทางวิศวกรรมประกอบด้วย 5 ขั้นตอน คือ

1. กำหนดปัญหา
2. วิเคราะห์ปัญหา
3. ค้นหาวิธีแก้ไข
4. ตัดสินใจเลือกแนวทางแก้ปัญหาที่เหมาะสม
5. ระบุวิธีแก้ปัญหา

1.5.2 ความท้าทายของการปรับเปลี่ยนสู่ดิจิทัลในวิศวกรรมซอฟต์แวร์

ความท้าทายของการปรับเปลี่ยนสู่ดิจิทัลในวิศวกรรมซอฟต์แวร์มีทั้งด้านเทคนิคและองค์กร [9] กล่าวคือ ในด้านเทคนิคนั้นมีการเปลี่ยนแปลงของเทคโนโลยีอย่างมากทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ รวมถึงการเปลี่ยนแปลงพฤติกรรมของผู้ใช้ ในด้านองค์กรนั้น การพัฒนาซอฟต์แวร์ต้องใช้ทรัพยากรมนุษย์เป็นหลักซึ่งเป็นทั้งศาสตร์และศิลป์ซึ่งพอจะสรุปได้ดังนี้

- **ต้องลดเวลาการเผยแพร่ซอฟต์แวร์สู่ตลาดมากกว่าที่เคยเป็นมา**

ด้วยนวัตกรรมใหม่ๆ และความคาดหวังจากลูกค้าที่ต้องการใช้บริการโดยเร็วที่สุด เนื่องจากการปรับเปลี่ยนสู่ดิจิทัลที่เพิ่มขึ้นในสังคม และเนื่องจากการมุ่งเน้นที่ผู้ใช้บริการ ดังนั้น เวลาในการเผยแพร่ซอฟต์แวร์ออกสู่ตลาดจึงต้องลดลงอีก ซึ่งเกิดขึ้นกับซอฟต์แวร์ที่พัฒนาขึ้นใหม่ทั้งหมด และคุณสมบัติใหม่สำหรับซอฟต์แวร์ที่มีอยู่แล้ว ส่งผลให้นักพัฒนาซอฟต์แวร์ต้องเตรียมกระบวนการพัฒนาในลักษณะที่ช่วยให้สามารถใช้งานฟังก์ชันใหม่ๆ ได้อย่างต่อเนื่อง ซึ่งการทำงานแบบอไจล์ หรือการพัฒนาโปรแกรมแบบเอ็กซ์ตรีมเป็นแนวทางในการเพิ่มความคล่องตัวซึ่งอธิบายไว้ในบทที่ 5

- **ความคล่องตัวและความรวดเร็วในการพัฒนาซอฟต์แวร์**

เนื่องจากมีเวลาในการออกสู่ตลาดที่ลดลง ผู้พัฒนาจึงต้องการระบบซอฟต์แวร์ที่ยืดหยุ่นพอที่จะพิจารณาข้อกำหนดใหม่ (Specification) ในภายหลัง นอกจากนี้ แรงกดดันจากการแข่งขันที่สูง ทำให้ต้องปรับเปลี่ยนคุณลักษณะของซอฟต์แวร์ที่วางแผนไว้แต่แรกด้วยคุณสมบัติอื่นๆ ดังนั้น กระบวนการพัฒนาจะต้องคล่องตัวและรวดเร็ว เพื่อตอบสนองความต้องการที่เปลี่ยนแปลงไป วิธีการพัฒนาซอฟต์แวร์แบบดั้งเดิม เช่น การใช้วงจรการพัฒนาระบบแบบน้ำตก (Water Fall Model) อาจจะไม่เหมาะสมกับสถานการณ์ จึงต้องการทำงานแบบอไจล์ (Agile) ซึ่งทำงานได้อย่างคล่องตัวและรวดเร็ว เพื่อให้มีความยืดหยุ่นเพียงพอที่จะพัฒนาคุณสมบัติใหม่ๆ เฉพาะที่จำเป็น มากกว่าการรอให้พัฒนาคุณสมบัติให้ครบถ้วนก่อนที่จะนำซอฟต์แวร์ออกสู่ตลาด ได้อธิบายไว้ในบทที่ 5

- **แนวคิดใหม่**

ในช่วงไม่กี่ปีที่ผ่านมามีแนวคิดใหม่ เช่น ข้อมูลขนาดใหญ่ (Big Data) อินเทอร์เน็ตของสรรพสิ่ง (Internet of Things, IoT) ประเทศไทย 4.0 (Thailand 4.0) อุตสาหกรรม 4.0 (Industry 4.0) และปัญญาประดิษฐ์ (Artificial Intelligence) เป็นต้น ดังนั้น นักพัฒนาซอฟต์แวร์จึงต้องเข้าใจข้อกำหนดและแนวคิดใหม่เหล่านี้ ซึ่งการเรียนรู้และการฝึกอบรมที่มีประสิทธิภาพเป็นสิ่งจำเป็นสำหรับนักพัฒนา ได้อธิบายไว้ในบทที่ 11



● อุปกรณ์และเทคโนโลยีใหม่

ด้วยการปรับเปลี่ยนสู่ดิจิทัลที่เพิ่มขึ้น อุปกรณ์ใหม่ๆ แบบพกพาจึงเป็นที่นิยม เช่น แว่นตาเสมือนจริง (VR Glasses, Virtual Reality) นาฬิกาอัจฉริยะ (Smart Watch) เป็นต้น รวมถึงอุปกรณ์อัจฉริยะทั้งหมดในบ้าน ซึ่งอุปกรณ์เหล่านี้ต้องการระบบนิเวศดิจิทัล (Digital Ecosystem) เช่น ระบบเครือข่ายไร้สาย ระบบปฏิบัติการ และซอฟต์แวร์ที่รองรับการทำงาน ดังนั้น นักพัฒนาซอฟต์แวร์จึงต้องเตรียมพร้อมสำหรับการพัฒนาซอฟต์แวร์ให้รองรับอุปกรณ์และเทคโนโลยีใหม่ทั้งหมด โดยจะต้องจัดการฝึกอบรม และให้อิสระแก่นักพัฒนาได้เรียนรู้แนวคิดใหม่ๆ ซึ่งเป็นปัจจัยสำคัญสู่ความสำเร็จ เนื่องจากอุปกรณ์และเทคโนโลยีใหม่ๆ จำเป็นต่อการแข่งขันในตลาด ได้อธิบายไว้ในบทที่ 6

● ความหลากหลายทางเทคโนโลยีดิจิทัล

อุปกรณ์และเทคโนโลยีดิจิทัลที่มีเป็นจำนวนมากส่งผลให้เกิดความท้าทาย อันเนื่องมาจากเวลาในการออกสู่ตลาดของอุปกรณ์ที่รวดเร็ว และมักจะใช้แพลตฟอร์มของตนเองซึ่งไม่สามารถใช้ร่วมกันผู้ผลิตอื่นๆ ได้ ทำให้ในโครงการเดียวกันอาจจะมีอุปกรณ์และเทคโนโลยีที่แตกต่างกัน นักพัฒนาจึงต้องหาวิธีแก้ปัญหาเพื่อจัดการกับความหลากหลายเหล่านี้ ในหนังสือนี้ได้อธิบายแนวทางในการต่อประสาน (Interface) ระหว่างเทคโนโลยีที่แตกต่างกันไว้ในบทที่ 10

● ระบบเครือข่ายที่ดีขึ้น

ระบบซอฟต์แวร์ที่พัฒนาขึ้นในปัจจุบัน ส่วนใหญ่เชื่อมต่อกับระบบซอฟต์แวร์อื่นที่มีอยู่แล้วบนระบบเครือข่าย โดยการใช้การต่อประสานเพื่อแลกเปลี่ยนข้อมูล เช่น การใช้เว็บเซอร์วิส (Web Services) การพัฒนาระบบซอฟต์แวร์จึงต้องเปลี่ยนจากการพัฒนาซอฟต์แวร์ทั้งระบบเป็นการพัฒนาเป็นส่วนประกอบ (Component) นักพัฒนาซอฟต์แวร์จึงจำเป็นต้องทราบหลักวิศวกรรมซอฟต์แวร์เชิงเซอร์วิสในบทที่ 10

● การให้บริการเสริม

ผู้ให้บริการมีการนำเสนอโครงสร้างพื้นฐาน แพลตฟอร์ม และซอฟต์แวร์เป็นแบบเซอร์วิส (Services) มากขึ้นเรื่อยๆ เมื่อเทียบกับการพัฒนาซอฟต์แวร์แบบเดิมซึ่งให้บริการแบบองค์รวม (Solution) แนวโน้มคือการนำฟังก์ชันที่มีอยู่มาใช้ใหม่ (Reused) แทนการพัฒนาตั้งแต่เริ่มต้น ดังนั้น วิธีการพัฒนาระบบซอฟต์แวร์จึงเปลี่ยนจากการพัฒนาทั้งระบบไปเป็นการพัฒนาแบบส่วนๆ แทน ซึ่งจะได้อภิปรายในการวิเคราะห์และออกแบบเชิงวัตถุ และการใช้ภาษาการสร้างแบบจำลองแบบครบวงจรในบทที่ 3 และ 4

● ตลาดเป็นของผู้ใช้

ทุกวันนี้มีการใช้ระบบซอฟต์แวร์มากขึ้นเรื่อยๆ บนสมาร์ทโฟนและแท็บเล็ต การวิเคราะห์ความต้องการของผู้ใช้จึงเปลี่ยนจากการวิเคราะห์แบบเดิมคือ แบบฟังก์ชัน (Functional) ไปเป็นการวิเคราะห์จากประสบการณ์ผู้ใช้ (Customer Journey) ซึ่งหมายความว่า นักพัฒนาซอฟต์แวร์ต้องให้ความสำคัญกับผู้ใช้ระบบมากกว่าที่เคย ซึ่งได้อธิบายกระบวนการนี้ไว้ในบทที่ 1

- **การขับเคลื่อนธุรกิจ**

แนวคิดหลักประการหนึ่งของการปรับเปลี่ยนสู่ดิจิทัลคือ การใช้ประโยชน์จากการแปลงเป็นดิจิทัล (Digitization) ที่เพิ่มขึ้นในสังคมเพื่อสร้างโมเดลธุรกิจใหม่ (Business Model) การสร้างนวัตกรรม (Innovation) บริการ (Services) หรือผลิตภัณฑ์ใหม่ๆ มีความสำคัญต่อการเติบโตอย่างยั่งยืน ความสามารถในการแข่งขันและการยกระดับคุณค่า เนื่องจากนวัตกรรมต้องการความรู้ทางเทคโนโลยี การสร้างโมเดลทางธุรกิจจึงไม่ใช่งานทางธุรกิจเพียงอย่างเดียว ดังนั้น นักพัฒนาซอฟต์แวร์จึงต้องคิดเชิงธุรกิจและคิดนอกกรอบมากขึ้น ฝ่ายบริหารจำเป็นต้องสนับสนุนนักพัฒนาให้มีอิสระในการทดลองใช้ความคิดใหม่ๆ และเปิดรับแนวคิดใหม่ๆ จากนักพัฒนา ดังรายละเอียดในบทที่ 1

- **ผสมผสานกับเทคโนโลยีแบบดั้งเดิม**

เนื่องจากอุปกรณ์และเทคโนโลยีใหม่ๆ เกิดขึ้นอย่างรวดเร็ว ส่งผลให้ระบบเทคโนโลยีดิจิทัลที่ใช้งานอยู่ล้าสมัยเร็วขึ้น แต่ด้วยข้อจำกัดด้านงบประมาณและทรัพยากร จึงไม่สามารถนำระบบใหม่มาทดแทนระบบเดิมได้ในทันที ดังนั้น นักพัฒนาซอฟต์แวร์จึงต้องสามารถบูรณาการระบบซอฟต์แวร์ใหม่ เข้ากับระบบเทคโนโลยีสารสนเทศและการสื่อสารที่ใช้งานอยู่ให้เกิดประโยชน์สูงสุด ซึ่งจะได้อธิบายไว้ในบทที่ 6 และ 10

1.5.3 แนวทางการเอาชนะความท้าทายของการปรับเปลี่ยนสู่ดิจิทัลในด้านวิศวกรรมซอฟต์แวร์

การเอาชนะความท้าทายของการปรับเปลี่ยนสู่ดิจิทัลในด้านวิศวกรรมซอฟต์แวร์ ในหนังสือเล่มนี้ได้อธิบายแนวทางให้นักพัฒนาซอฟต์แวร์เตรียมพร้อมสำหรับโครงการซอฟต์แวร์ในอนาคต และเพื่อให้สามารถแข่งขันและรองรับความต้องการของผู้ใช้ที่เพิ่มขึ้นอย่างมากมายได้

- **ไมโครเซอร์วิส (Microservices)** เป็นบริการขนาดเล็กและเป็นอิสระที่ทำงานร่วมกันผ่านเว็บ แนวคิดนี้สามารถลดเวลาในการออกสู่ตลาดเมื่อพัฒนาฟังก์ชันหรือบริการ (Services) ใหม่ๆ โดยแต่ละบริการสามารถพัฒนาและปรับใช้แยกกันได้ โดยไม่ต้องมีการประสานงานกับทีมพัฒนาต่างๆ และสามารถใช้ในเทคโนโลยีที่ต่างกันในแต่ละไมโครเซอร์วิสตามความต้องการ แทนที่จะใช้มาตรฐานเดียวกันทั้งหมด ตัวอย่างเช่น บริการหนึ่งสามารถเขียนด้วยภาษาจาวา (Java) และอีกหนึ่งบริการในภาษา C# เป็นต้น นอกจากนี้ ยังสามารถนำกลับมาใช้ใหม่ได้ ดังรายละเอียดในบทที่ 10
- **การพัฒนาเชิงเซอร์วิส** เป็นการพัฒนาโดยใช้แนวคิดเชิงบริการ (Services Oriented) ซึ่งสามารถบูรณาการกับระบบอื่นทั้งภายในและภายนอกหน่วยงานได้ เช่น การเชื่อมต่อกับเครือข่ายสังคม เช่น Facebook, YouTube เป็นต้น แต่การพัฒนาในลักษณะนี้ต้องการวิสัยทัศน์ และวัตถุประสงค์ทางธุรกิจที่ชัดเจน นอกจากมุมมองทางธุรกิจแล้ว ยังต้องมีแนวปฏิบัติที่ดีในการรักษาความมั่นคงปลอดภัยของระบบสารสนเทศควบคู่กันไปด้วย ซึ่งอธิบายไว้ในบทที่ 10



- **ระบบอัตโนมัติ** สามารถลดเวลาที่เกิดจากงานที่เกิดซ้ำๆ ได้อย่างมาก ซึ่งจะเป็นการเพิ่มประสิทธิภาพการทำงานเนื่องจากไม่ต้องจัดการกับงานเหล่านี้อีกต่อไป เช่น การใช้บริการคลาวด์ช่วยลดความซับซ้อนและลดเวลาสำหรับการปรับใช้ หรือการกำหนดค่าโครงสร้างพื้นฐาน เป็นต้น ดังแสดงไว้ในบทที่ 8 และบทที่ 9
- **ยึดหลักธรรมาภิบาลทางเทคนิค** การกำกับดูแลด้านเทคนิคเป็นแนวทางที่สำคัญ ซึ่งทีมพัฒนาทั้งหมดต้องปฏิบัติตามเพื่อให้เกิดความเป็นเอกภาพในการพัฒนา แม้ว่าผู้พัฒนาจะต้องการการออกแบบหรือการตัดสินใจทางเทคโนโลยีเป็นของตนเอง แต่ก็ควรกำหนดมาตรฐานหรือแนวทางปฏิบัติทั่วทั้งองค์กรในการพัฒนาซอฟต์แวร์หรือบริการ เนื่องจากแนวทางดังกล่าวเป็นการวางรากฐานของการบูรณาการ เช่น การใช้หลักธรรมาภิบาลข้อมูล (Data Governance) เป็นต้น นอกจากนี้ต้องหลีกเลี่ยงการตัดสินใจใดๆ ที่ส่งผลกระทบต่อทั้งระบบ ในหนังสือนี้ได้กล่าวไว้ในบทที่ 6
- **การเลือกใช้เทคโนโลยีดิจิทัลที่เหมาะสม** การเติบโตอย่างรวดเร็วของเทคโนโลยีใหม่ๆ ทำให้ผู้พัฒนาต้องใช้ความพยายามในการเรียนรู้และฝึกอบรม สร้างความยุ่งยากในการเลือกใช้เทคโนโลยีที่เหมาะสม หลายกรณีที่ตัดสินใจดำเนินการไปแล้ว แต่อาจมีเทคโนโลยีใหม่ๆ เกิดขึ้น ดังนั้น ผู้พัฒนาจึงควรเปิดใจยอมรับเทคโนโลยีหรือแนวทางใหม่ๆ โดยไม่ยึดติดกับการตัดสินใจก่อนหน้านี้ ซึ่งเทคโนโลยีหรือแนวทางใหม่นี้ควรได้รับการวิเคราะห์โดยละเอียด
- **การออกแบบซอฟต์แวร์** การออกแบบคือความประทับใจแรกที่ได้รับจากซอฟต์แวร์ ซึ่งไม่ใช่แค่เพียงการออกแบบแบบวิธีการติดต่อกับผู้ใช้ (User Interface) ของแอปพลิเคชันเท่านั้น แต่รวมถึงการออกแบบวิธีการแก้ปัญหาที่สามารถนำไปใช้กับบริบทต่างๆ อีกด้วย เช่น วิธีเชื่อมต่อของแอปพลิเคชันหรือบริการ (Application Program Interface, API) ต้องออกแบบ API ให้เข้าใจง่าย ผู้พัฒนาจึงควรเน้นการออกแบบที่ดีสำหรับผู้ใช้ แทนที่จะเน้นเฉพาะด้านฟังก์ชันการทำงานของซอฟต์แวร์แต่เพียงอย่างเดียว ดังแสดงเกี่ยวกับเรื่องนี้ในบทที่ 8 และบทที่ 9
- **วัฒนธรรมการทำงานร่วมกัน** การทำงานร่วมกันเป็นทีมสามารถส่งผลดีต่อประสิทธิภาพการทำงาน และความตระหนักในตนเองของสมาชิกแต่ละคน แต่การสร้างวัฒนธรรมของทีมที่ได้อาจเป็นเรื่องยาก เนื่องจากมีหลายปัจจัยที่ต้องพิจารณา เช่น ความแตกต่างทางความคิด ความรู้ ประสบการณ์ของสมาชิกในทีม เป็นต้น ดังนั้น ผู้บริหารจึงควรให้ความสำคัญกับสมาชิกในทีมพัฒนา เปิดกว้างสำหรับการอภิปรายและการคิดนอกกรอบ ส่งเสริมการเรียนรู้เทคนิคใหม่ๆ และจัดให้มีการฝึกอบรม ซึ่งได้กล่าวไว้ในบทที่ 11

1.6 คำนิยามและคำจำกัดความที่ใช้ในวิศวกรรมซอฟต์แวร์

ซอฟต์แวร์ (Software)

โดยทั่วไปจะหมายถึง โปรแกรมคอมพิวเตอร์ และคู่มือที่เกี่ยวข้องอาจจะพัฒนาขึ้นเพื่อใช้กับลูกค้าทั่วไป ซึ่งเรียกว่า “ผลิตภัณฑ์ทั่วไป” (Generic Products) และที่ปรับให้ใช้กับเฉพาะรายที่เรียกว่า “ผลิตภัณฑ์ปรับแต่ง” (Customized Products) ซึ่งพัฒนาเฉพาะสำหรับลูกค้ารายใดรายหนึ่งตามความต้องการที่เฉพาะเจาะจง

วิศวกรรมซอฟต์แวร์ (Software Engineering)

คือ การประยุกต์ใช้หลักการทางวิศวกรรมเพื่อใช้ในการพัฒนาซอฟต์แวร์ ซึ่งเป็นการบริหารจัดการอย่างเป็นระบบ และใช้เครื่องมือต่างๆ ที่เหมาะสมตามความต้องการของลูกค้าภายใต้ข้อจำกัดของทรัพยากรและเวลา

ความแตกต่างระหว่าง วิศวกรรมซอฟต์แวร์ (Software Engineering) กับวิทยาการคอมพิวเตอร์ (Computer Science)

วิทยาการคอมพิวเตอร์จะให้ความสำคัญกับทฤษฎีและพื้นฐานของคอมพิวเตอร์ ในขณะที่วิศวกรรมซอฟต์แวร์ให้ความสำคัญกับการพัฒนาซอฟต์แวร์ และการส่งมอบซอฟต์แวร์ที่ถูกต้องตรงตามความต้องการของผู้ใช้ ซึ่งทฤษฎีทางวิทยาการคอมพิวเตอร์ไม่เพียงพอที่จะแทนวิทยาการด้านวิศวกรรมซอฟต์แวร์ได้

ความแตกต่างระหว่าง วิศวกรรมซอฟต์แวร์ (Software Engineering) กับวิศวกรรมระบบ (System Engineering)

วิศวกรรมระบบให้ความสำคัญเกี่ยวกับระบบของคอมพิวเตอร์ เช่น ฮาร์ดแวร์ (Hardware) ระบบปฏิบัติการ (Operating System) ซอฟต์แวร์ระบบ (System Software) หรือระบบเครือข่าย (Networking) ส่วนวิศวกรรมซอฟต์แวร์จะเน้นเกี่ยวกับข้อกำหนดความต้องการด้านซอฟต์แวร์ การออกแบบสถาปัตยกรรม การรวบรวมและการส่งมอบซอฟต์แวร์

กระบวนการทางซอฟต์แวร์ (Software Process)

คือ ชุดของกิจกรรมที่ใช้ในการพัฒนา หรือรองรับการเปลี่ยนแปลงของซอฟต์แวร์ ประกอบด้วย 4 กิจกรรม

1. **ข้อกำหนดของซอฟต์แวร์ (Specification)** เป็นการระบุฟังก์ชันการทำงานของระบบที่ต้องการจะพัฒนา
2. **การพัฒนา (Development)** ระบุถึงรายละเอียดและขั้นตอนในการพัฒนาระบบซอฟต์แวร์ตามข้อกำหนด
3. **การตรวจสอบ ประเมินผล (Validation)** เป็นการระบุถึงวิธีการตรวจสอบการทำงานของระบบที่พัฒนาขึ้น รวมทั้งวิธีการในการประเมินผลความถูกต้องของระบบ และความพึงพอใจของผู้ใช้ระบบ
4. **การรองรับการเปลี่ยนแปลงที่อาจเกิดขึ้น (Evolution)** โดยทั่วไป ระบบที่พัฒนาขึ้นจะสามารถรองรับงานได้ในระยะเวลาหนึ่ง และจะต้องมีวิวัฒนาการ เนื่องจากความต้องการเปลี่ยนแปลงไป หรือเทคโนโลยีมีความเจริญก้าวหน้ามากขึ้นในอนาคต



แบบจำลองกระบวนการทำงานของซอฟต์แวร์ (Software Process Model)

คือ ตัวแบบที่ใช้เป็นตัวแทนของกระบวนการทำงานของซอฟต์แวร์ เพื่ออธิบายขั้นตอนการทำงานของระบบเดิม และระบบที่จะออกแบบใหม่ ซึ่งจะทำให้ผู้พัฒนาสามารถเข้าใจและพัฒนาระบบซอฟต์แวร์ให้ตรงตามความต้องการของผู้ใช้ได้ ซึ่งสามารถนำเสนอได้หลายรูปแบบ เช่น

- มุมมองการลำดับขั้นตอนของกิจกรรม (Workflow Perspective)
- มุมมองการไหลของสารสนเทศ (Data-Flow Perspective)
- ภาษายูเอ็มแอล (Unifies Modeling Language, UML)
- มุมมองด้านเงื่อนไขหรือกฎที่เกี่ยวข้อง (Role Action Perspective)

การปรับรื้อกระบวนการทางธุรกิจ (Business Process Re-Engineering, BPR) กับการปรับปรุงกระบวนการทางธุรกิจ (Business Process Improvement, BPI)

ในองค์กรสมัยใหม่ที่มีการปรับเปลี่ยนขั้นตอนการทำงานให้มีประสิทธิภาพที่เรียกว่า “การปรับรื้อกระบวนการทางธุรกิจ” (BPR) และ “การปรับปรุงกระบวนการทางธุรกิจ” (BPI) ในทางปฏิบัติมีความจำเป็นจะต้องทำ BPR หรือ BPI ให้เสร็จก่อน แล้วจึงจะทำแบบจำลองกระบวนการทำงานของซอฟต์แวร์ ทั้งนี้ เพื่อให้ระบบซอฟต์แวร์ตรงกับขั้นตอนการทำงานใหม่หลังจากที่มีการปรับเปลี่ยนขั้นตอนการทำงานแล้ว

ต้นทุนของซอฟต์แวร์ (Software Cost)

ต้นทุนด้านซอฟต์แวร์จะมีมูลค่าสูงกว่าองค์ประกอบอื่นๆ ในการพัฒนาระบบ ซึ่งมีมูลค่าสูงกว่าฮาร์ดแวร์เสมอ และมีมูลค่าสูงขึ้นตลอดเวลา ซึ่งตรงข้ามกับฮาร์ดแวร์ซึ่งมีราคาถูกลงเรื่อยๆ นอกจากนี้ ค่าใช้จ่ายในการบำรุงรักษาซอฟต์แวร์ บางครั้งสูงกว่าการพัฒนาซอฟต์แวร์ วิศวกรรมซอฟต์แวร์จึงคำนึงถึงการควบคุมค่าใช้จ่ายในการผลิตซอฟต์แวร์ให้เหมาะสม ทั้งนี้ เพื่อลดค่าใช้จ่ายในการพัฒนาและบำรุงรักษาซอฟต์แวร์ โดยที่ระบบยังคงทำงานได้อย่างมีประสิทธิภาพ

ต้นทุนของการพัฒนาซอฟต์แวร์ (Cost of Software Development)

- ประมาณ 60% ของต้นทุนเป็นค่าใช้จ่ายในการพัฒนา 40% เป็นค่าทดสอบ ซึ่งถ้าเป็นการพัฒนาซอฟต์แวร์เฉพาะทาง (Customized Software) ค่าใช้จ่ายในการทดสอบจะสูงกว่าค่าใช้จ่ายในการพัฒนา เพราะต้องปรับเปลี่ยนให้ตรงกับความต้องการของลูกค้า
- ค่าใช้จ่ายจะขึ้นกับระบบการพัฒนา และความต้องการ รวมถึงประสิทธิภาพและความน่าเชื่อถือของระบบ
- ค่าใช้จ่ายยังขึ้นอยู่กับรูปแบบของการพัฒนาซอฟต์แวร์ด้วย เช่น กรณีที่มีการทำงานผ่านระบบเครือข่าย ค่าใช้จ่ายก็จะสูงขึ้น

วิธีการวิศวกรรมซอฟต์แวร์ (Software Engineering Methods)

1. **Structured Approaches to Software Development** มีโครงสร้างเพื่อให้ง่ายต่อการพัฒนาซอฟต์แวร์ รวมถึงการสร้างแบบจำลองของระบบ บันทึกรหัส กฎเกณฑ์ คำแนะนำในการออกแบบ และแนวทางในการดำเนินการ
2. **Model Descriptions** คำอธิบายแบบจำลองโดยแสดงเป็นสัญลักษณ์ ซึ่งควรมีการจัดเตรียมไว้เพื่อเป็นแนวทางในการพัฒนา

3. **Rules** กฎเกณฑ์หรือเงื่อนไขที่ใช้กับแบบจำลองของระบบ

4. **Recommendations** เป็นคำแนะนำเพื่อการออกแบบในทางปฏิบัติ

5. **Process Guidance** แนวทางในการดำเนินการระบุกิจกรรมต่างๆ ที่ต้องทำตามลำดับก่อนหลัง

ลักษณะของซอฟต์แวร์ที่ดี (Attributes of Good Software)

ซอฟต์แวร์ที่ดีควรจะส่งมอบด้วยคุณสมบัติที่ถูกต้องตามความต้องการของผู้ใช้ ถูกต้องและมีประสิทธิภาพที่ดี ง่ายต่อการบำรุงรักษา มีรายละเอียดดังนี้

- **Acceptability/Usability** ซอฟต์แวร์ต้องใช้งานได้จริง และเข้ากันได้กับระบบอื่นๆ ที่ผู้ใช้มีการใช้งานอยู่
- **Dependability and Security** ซอฟต์แวร์ต้องเชื่อถือได้โดยมีการทำงานอย่างถูกต้องและปลอดภัย
- **Efficiency** ซอฟต์แวร์ควรใช้ทรัพยากรของระบบอย่างมีประสิทธิภาพ รวมถึงมีการทำงานที่รวดเร็ว
- **Maintainability** การพัฒนาซอฟต์แวร์อย่างยืดหยุ่น เพื่อตอบสนองความต้องการที่เปลี่ยนแปลงไปของลูกค้าตามสภาพแวดล้อมทางธุรกิจที่เปลี่ยนแปลงไป

ความท้าทายที่สำคัญที่วิศวกรรมซอฟต์แวร์ต้องเผชิญ (Key Challenges Facing Software Engineering)

ปัจจุบัน ระบบคอมพิวเตอร์มีจำนวนเพิ่มมากขึ้น และมีความซับซ้อนมากขึ้นด้วยเช่นกัน แต่เวลาในการส่งมอบซอฟต์แวร์กลับลดลง วิศวกรรมซอฟต์แวร์จึงเผชิญกับความท้าทายหลัก 3 ประการ ดังนี้

- **Legacy Systems Challenge** ระบบซอฟต์แวร์เดิมส่วนใหญ่ยังคงทำหน้าที่สำคัญทางธุรกิจ ความท้าทายจึงอยู่ที่การบำรุงรักษา และอัปเดตซอฟต์แวร์ให้มีค่าใช้จ่ายต่ำ โดยยังคงให้บริการได้อย่างราบรื่น และระบบใหม่ก็ต้องพัฒนาให้สามารถทำงานได้จริง และสามารถเชื่อมโยงหรือถ่ายโอนข้อมูลกับระบบเดิมที่ใช้อยู่ได้อีกด้วย
- **Heterogeneity Challenge** ระบบบนเครือข่ายจะเป็นแบบกระจายอันเกิดจากการผสมผสานของฮาร์ดแวร์ และซอฟต์แวร์ที่มีความหลากหลาย ความท้าทายจึงอยู่ที่การพัฒนาซอฟต์แวร์ที่มีความยืดหยุ่นเพียงพอที่จะรองรับกับความหลากหลายนี้ได้นั่นเอง
- **Delivery Challenge** เทคนิคทางวิศวกรรมซอฟต์แวร์ดั้งเดิมมักใช้เวลานานในการพัฒนาว่าจะได้ซอฟต์แวร์ที่มีคุณภาพ แต่ธุรกิจในปัจจุบันต้องตอบสนองอย่างรวดเร็วเพื่อให้ทันความเปลี่ยนแปลง ความท้าทายจึงอยู่ที่การลดเวลาในการส่งมอบซอฟต์แวร์ที่เป็นระบบขนาดใหญ่ และซับซ้อนโดยไม่กระทบต่อคุณภาพ

หลักจริยธรรมทางวิศวกรรมซอฟต์แวร์ (Software Engineering Ethics)

- หลักจริยธรรมทางวิศวกรรมซอฟต์แวร์จะให้ความสำคัญกับความรับผิดชอบต่อลูกค้ามากกว่าทางเทคนิค
- หลักจริยธรรมทางวิศวกรรมซอฟต์แวร์จะให้ความสำคัญกับความซื่อสัตย์ต่อลูกค้า ไม่นำความลับไปเผยแพร่
- หลักจริยธรรมมีความสำคัญมากกว่ากฎหมาย

การปรับเปลี่ยนสู่ดิจิทัลด้วยวิศวกรรมซอฟต์แวร์ (Digital Transformation with Software Engineering)

วิศวกรรมซอฟต์แวร์จะปรับเปลี่ยนวิธีทำงานให้อยู่ในรูปแบบดิจิทัล (Digitization) ตามหลักการวิศวกรรมซอฟต์แวร์ ดังขั้นตอนต่อไปนี้



เริ่มต้นจากการค้นหาและเข้าใจปัญหา ซึ่งถ้างานที่กำลังจะทำนั้นไม่มีปัญหา ก็ไม่มีความจำเป็นที่จะต้องไปวิเคราะห์ให้เสียเวลา เมื่อทราบปัญหาแล้วจึงเริ่มทำความเข้าใจปัญหานั้นๆ จากนั้น ประเมินความคุ้มค่าในการแก้ปัญหา ถ้าหากประเมินแล้วคุ้มค่าที่จะลงทุนในการพัฒนาระบบสารสนเทศ ซึ่งกำหนดความต้องการที่จะแก้ปัญหา หรือที่เรียกว่า “ข้อกำหนด” (Term Of Reference, TOR หรือ Specification) หลังจากนั้นจะนำ TOR นี้ไปให้กับผู้พัฒนาระบบเพื่อวิเคราะห์วิธีการแก้ปัญหา หรือเรียกว่า “ข้อเสนอโครงการ” (Proposal) ซึ่งอาจจะมีได้หลายวิธี และมีการประมวลเพื่อหาวิธีการแก้ปัญหาที่ดีที่สุด ซึ่งจะนำไปกำหนดรายละเอียดในเชิงเทคนิค จากนั้นจะดำเนินการพัฒนาตามวงจรการพัฒนา ระบบสารสนเทศ (System Development Life Cycle, SDLC) หรือแบบอไจล์ (Agile) ต่อไป ผลการติดตามและประเมินผลระบบสารสนเทศที่พัฒนาขึ้น จะทำให้ทราบถึงประสิทธิภาพของระบบสารสนเทศเพื่อนำไปปรับปรุงระบบต่อไป

แบบฝึกหัดท้ายบท

1. อธิบายความสำคัญของเศรษฐกิจดิจิทัล
2. เพราะเหตุใดเทคโนโลยีดิจิทัลจึงมีความสำคัญกับเศรษฐกิจดิจิทัล
3. อธิบายแนวทางในการปรับเปลี่ยนสู่ดิจิทัล
4. อธิบายการเดินทางของลูกค้า (Customer Journey) ในธุรกิจ e-Commerce
5. ซอฟต์แวร์ที่ดีจะต้องมีลักษณะเช่นใด
6. ต้นทุนของการพัฒนาซอฟต์แวร์เกิดจากส่วนใด
7. อะไรคือสาเหตุสำคัญที่ทำให้การพัฒนาซอฟต์แวร์ไม่ประสบผลสำเร็จ
8. การทำ Re-Engineering หรือ BPI มีผลอย่างไรในการพัฒนาระบบ
9. จงอธิบายความแตกต่างระหว่าง Software Engineering กับ Computer Science
10. เพราะเหตุใดจึงกล่าวว่า การพัฒนาซอฟต์แวร์ในปัจจุบันมีความยุ่งยาก ซับซ้อนกว่าในอดีต

เอกสารอ้างอิง

1. ธนชาติ นุ่มนนท์. กลยุทธ์สู่ความสำเร็จด้าน Digital Transformation จาก Framework ของ BCG [ออนไลน์]. 2019, แหล่งที่มา <https://thanachart.org/2019/02/19/แนวทางสู่การทำ-digital-transformation/> [1 ตุลาคม 2563].
2. ธนพงศ์พรพนธ์ ธีญญรัตน์ตกุล. Digital Transformation in Action เปลี่ยนธุรกิจในยุคดิจิทัล Step by Step. กรุงเทพฯ. U23 Academy, 2562.
3. ปานใจ ธารทัศนวงศ์. การวิเคราะห์และออกแบบระบบเทคโนโลยีสารสนเทศในมุมมองด้านการบริหาร. กรุงเทพฯ: สันทวีการพิมพ์, 2554.
4. พิรุวรรณ กิตติคุณ. ภาครัฐไทยกับการก้าวสู่รัฐบาลดิจิทัล. สำนักวิชาการ สำนักงานเลขาธิการสภาผู้แทนราษฎร. 2559.

5. พรศักดิ์ อรุณชัยรัตน์. เศรษฐกิจดิจิทัล. สำนักพิมพ์แมคกรอ-ฮิล อินเทอร์เน็ตชั้นแนล เอ็นเตอร์ไพรส์ แอลแอลซี, 2559.
6. พระราชบัญญัติการบริหารงานและการให้บริการภาครัฐผ่านระบบดิจิทัล พุทธศักราช 2562. (2562, 22 พฤษภาคม). ราชกิจจานุเบกษา. หน้า 57-66.
7. สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน). (ร่าง) แผนพัฒนารัฐบาลดิจิทัลของประเทศไทย พ.ศ. 2563-2565. 2562.
8. EARLEY Information Science, Inc. (2020) Building a Successful Digital Transformation Roadmap [Online], available <https://info.earley.com/building-successful-digital-transformation-roadmap-whitepaper> [2020, October 1].
9. Michael Gebhart, Pascal Giessler and Sebastian Abeck. Challenges of the Digital Transformation in Software Engineering. Proceeding of The Eleventh International Conference on Software Engineering Advances (ICSEA2016). Rome, Italy. 21-25 August 2016.
10. Sommerville, Ian. Software Engineering. 8th ed. U.S.A.: Pearson Prentice Hall, 2007.
11. Sommerville, Ian. Software Engineering. 9th ed. U.S.A.: Pearson Prentice Hall, 2011.
12. Sommerville, Ian. Software Engineering. 10th ed. U.S.A.: Pearson Prentice Hall, 2016.
13. World Economic Forum. (2020). Four Themes of Becoming a Digital Enterprise [Online]. Available <http://reports.weforum.org/digital-transformation/becoming-a-digital-enterprise> [2020, October 1].

CHAPTER

2

การวิเคราะห์เชิงโครงสร้าง STRUCTURED ANALYSIS

วัตถุประสงค์ (Objectives)

- เพื่ออธิบายแนวคิดเชิงโครงสร้าง
- เพื่ออธิบายการวิเคราะห์เชิงโครงสร้าง
- เพื่ออธิบายแผนภาพการไหลข้อมูล (Data Flow Diagram, DFD)

เนื้อหาประจำบท (Content)

- 2.1 การวิเคราะห์เชิงโครงสร้าง (Structure Analysis and Design)
- 2.2 แผนภาพการไหลของข้อมูล (Data Flow Diagram)
- 2.3 แผนภาพการไหลของข้อมูลกับการปรับโครงสร้างทางธุรกิจ



System Development Life Cycle หรือ SDLC Process

2.1 การวิเคราะห์เชิงโครงสร้าง (Structure Analysis And Design)

การวิเคราะห์เชิงโครงสร้างได้รับการพัฒนาและเริ่มใช้ในช่วงปี ค.ศ. 1969-1973 โดย Douglas T. Ross และ SofTech, Inc. [7] วิธีการนี้เป็นส่วนหนึ่งของชุดวิธีการที่มีโครงสร้าง ซึ่งแสดงถึงการวิเคราะห์การออกแบบ และเทคนิคการเขียนโปรแกรมที่ได้รับการพัฒนาขึ้นเพื่อตอบสนองต่อปัญหาที่โลกซอฟต์แวร์เผชิญตั้งแต่ปี ค.ศ. 1960-1980 โดยในเวลานั้นการเขียนโปรแกรมเชิงพาณิชย์ส่วนใหญ่ทำในภาษาโคบอล (COBOL) ฟอ์แทรน (FORTRAN) ตามด้วยภาษาปาสคาล (Pascal) ภาษาเบสิก (BASIC) และภาษาซี (C) แต่เนื่องจากวิธีการนี้ไม่มีเทคนิคมาตรฐานในการจัดทำในการวิเคราะห์และการออกแบบระบบ ทำให้การพัฒนาระบบสารสนเทศทำได้ยากขึ้นเรื่อยๆ เมื่อระบบมีขนาดใหญ่ขึ้นและซับซ้อนมากขึ้น อย่างไรก็ตามวิธีการนี้ยังมีใช้อยู่ในปัจจุบันในกรณีที่พัฒนาระบบด้วยภาษาเชิงโครงสร้าง เช่น การใช้ภาษาซีในการพัฒนาระบบอินเทอร์เน็ตประสาสนเทศ (Internet of Things, IoT) เป็นต้น

การวิเคราะห์เชิงโครงสร้าง (Structure Analysis and Design) เป็นการวิเคราะห์การทำงานตามหน้าที่ (Function) ของแต่ละส่วนงานหรือแต่ละระบบย่อย เพื่ออธิบายการนำเข้าข้อมูล กระบวนการทำงาน และการแสดงผล รวมทั้งปฏิสัมพันธ์กับผู้ที่เกี่ยวข้อง เครื่องมือที่ช่วยในการวิเคราะห์ระบบด้วยวิธีนี้คือ แผนภาพการไหลของข้อมูล (Data Flow Diagram)

2.2 แผนภาพการไหลของข้อมูล (Data Flow Diagram)

แผนภาพการไหลของข้อมูล (Data Flow Diagram, DFD) [1] เป็นเครื่องมือที่ใช้ในการวิเคราะห์กระบวนการทำงานของผู้ใช้เพื่อให้ผู้พัฒนาระบบ และผู้ใช้ระบบเห็นภาพเดียวกัน ซึ่งเป็นการตรวจสอบว่าผู้พัฒนาระบบเข้าใจกระบวนการทำงานอย่างถูกต้อง และนำไปสู่การออกแบบและดำเนินการที่ครบถ้วนสมบูรณ์ต่อไป

แผนภาพการไหลของข้อมูลประกอบด้วยสัญลักษณ์ แสดงในรูปที่ 2.1 ต่อไปนี้

- กระบวนการ (Process) เป็นขั้นตอนการทำงานของหน่วยงาน
- แหล่งเก็บข้อมูล (Data Store) แสดงที่เก็บหรือที่พักของข้อมูลในระบบที่กำลังวิเคราะห์
- จุดเริ่มต้น (Source) หรือจุดจบ (Sink) อาจจะเป็นคนหรือระบบที่อยู่ภายนอกและมีความเชื่อมโยงกับระบบ ซึ่งอาจจะเป็นส่วนที่เริ่มต้นของการกำเนิดข้อมูล หรือเป็นส่วนที่รับข้อมูลจากระบบ
- การไหลของข้อมูลหรือกระแสข้อมูล (Data Flow) แสดงทิศทางการไหลของข้อมูลระหว่างกระบวนการทำงาน แหล่งเก็บข้อมูล จุดเริ่มต้นหรือจุดสุดท้ายของระบบ



รูปที่ 2.1 สัญลักษณ์ของแผนภาพการไหลของข้อมูลแบบ Gane และ Sarson [2]

2.2.1 แผนภาพบริบท (Context Diagram)

เป็นส่วนหนึ่งของแผนภาพการไหลของข้อมูล ซึ่งเป็นแผนภาพที่อธิบายภาพรวมขององค์กรที่แสดงขอบเขตของระบบกับองค์ประกอบภายนอก (External Entity) ซึ่งอาจจะเป็นคน สัตว์ สิ่งของ หรือระบบอื่นๆ ที่อยู่นอกระบบ และมีการติดต่อกับระบบนั้น โดยแสดงกระแสข้อมูลที่สำคัญระหว่างองค์ประกอบภายนอกและระบบ สัญลักษณ์ที่ใช้จะแสดงกระบวนการทำงานหลักเพียงหนึ่งเดียว และไม่แสดงกระบวนการ (Process) ย่อย และแหล่งจัดเก็บข้อมูล ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 แผนภาพบริบท (Context Diagram) [1]

รูปที่ 2.2 ลูกค้า ห้องครัว และผู้จัดการร้าน เป็น Entity ซึ่งลูกค้าจะเป็นจุดเริ่มต้น (Source) ของการส่งข้อมูลอาหารที่ต้องการจะสั่งไปยังระบบสั่งอาหาร จากนั้น ระบบจะรวบรวมคำสั่งซื้อ และส่งรายการอาหารที่ลูกค้าสั่งซื้อไปยังห้องครัวซึ่งเป็นจุดสุดท้าย (Sink) ที่รับข้อมูลจากระบบ เมื่อห้องครัวทำอาหารเสร็จก็จะนำไปส่งให้ลูกค้าและเก็บเงินค่าอาหาร และเมื่อได้รับชำระเงินแล้ว ระบบก็จะพิมพ์ใบเสร็จรับเงินให้กับลูกค้า ในกรณีนี้ลูกค้าจะเป็นทั้งจุดเริ่มต้น (Source) หรือจุดจบ (Sink) ในคนเดียวกัน เมื่อสิ้นวัน ระบบก็จะทำการพิมพ์รายงานสรุปลงขายให้กับผู้จัดการร้าน ซึ่งเป็นจุดสิ้นสุดของกระแสข้อมูลในระบบนี้

2.2.2 แผนภาพการไหลของข้อมูลระดับที่ 0 (DFD Level 0)

เป็นแผนภาพที่แสดงรายละเอียดที่มากกว่าแผนภาพบริบท (Context Diagram) โดยที่แผนภาพในระดับนี้จะแสดงกระบวนการหลัก กระแสการไหลของข้อมูล และแหล่งเก็บข้อมูล กระบวนการในระดับนี้จะระบุหมายเลขเป็น 1.0, 2.0 และ 3.0 เป็นต้น



รูปที่ 2.3 แผนภาพการไหลของข้อมูลระดับที่ 0 [1]

รูปที่ 2.3 แสดงรายละเอียดที่มากกว่าแผนภาพบริบท (Context Diagram) โดยแสดงส่วนประกอบทั้งหมดคือ กระบวนการ (Process) จำนวน 4 ขั้นตอน แหล่งเก็บข้อมูล (Data Store) จำนวน 2 แห่ง และ Entity คือ ลูกค้า ห้องครัว และผู้จัดการร้าน ซึ่งจะสังเกตได้ว่า จำนวน Entity ในแผนภาพบริบทจะต้องตรงกับในแผนภาพการไหลของข้อมูล (DFD)

นอกจากนี้ จำนวนกระแสข้อมูลในแผนภาพทั้งสองแบบต้องตรงกันด้วย เมื่อลูกค้าส่งคำสั่งซื้อมาที่กระบวนการที่ 1.0 ซึ่งทำหน้าที่รับข้อมูลและแปลงเป็นรายการอาหารแล้วส่งให้กับห้องครัว ในขณะเดียวกันจากกระบวนการที่ 1.0 เมื่อพ่อครัวทำอาหารเสร็จและส่งให้ลูกค้า ลูกค้าก็จะชำระเงิน กระบวนการนี้ก็จะพิมพ์ใบเสร็จรับเงินให้ลูกค้า พร้อมกันนั้นก็ส่งข้อมูลสินค้าที่ขายได้ไปยังกระบวนการที่ 2.0 คือ การปรับปรุงฐานข้อมูลการขายสินค้า กระบวนการนี้จะจัดรูปแบบของข้อมูลรายการขายสินค้าเพื่อให้ตรงกันกับโครงสร้างของแฟ้มข้อมูลการขายสินค้า

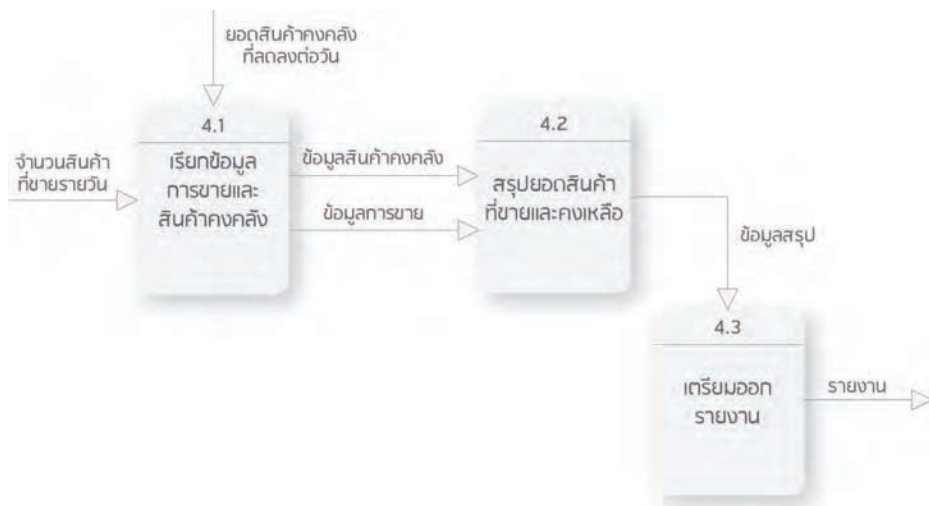
นอกจากนั้น กระบวนการที่ 1.0 ยังส่งข้อมูลสินค้าคงคลังไปยังกระบวนการที่ 3.0 ทำหน้าที่ปรับปรุงแฟ้มข้อมูลสินค้าคงคลังด้วยข้อมูลที่จัดรูปแบบให้ตรงกับโครงสร้างที่กำหนดไว้ เมื่อสิ้นวัน กระบวนการที่ 4.0 จะทำการออกรายงานสรุปยอด โดยเรียกข้อมูลรวมประจำวันจากฐานข้อมูลการขายสินค้าและสินค้าคงคลัง เพื่อทำการประมวลผลและพิมพ์รายงานสรุปให้กับผู้จัดการร้านต่อไป

2.2.3 การแยกองค์ประกอบ (Decomposition)

การแยกองค์ประกอบเป็นกระบวนการวนซ้ำ เพื่อให้คำอธิบายของระบบมีรายละเอียดปลีกย่อยเพิ่มขึ้น เปรียบเหมือนกับการมองวัตถุจากที่สูงซึ่งจะเห็นภาพขนาดเล็ก และเมื่อลดความสูงลงจะทำให้เห็นภาพใหญ่และชัดเจนขึ้น ด้วยวิธีการนี้ทำให้สามารถขยายภาพการทำงานของกระบวนการที่ต้องการรายละเอียด โดยจะจำแนกรายละเอียดเป็นตัวเลขในลำดับที่สูงขึ้น เช่น กระบวนการหมายเลข 4.0 เมื่อมีการแยกองค์ประกอบจะเป็น 4.1, 4.2, 4.3 เป็นต้น และเมื่อนำกระบวนการ 4.1 มาแยกองค์ประกอบจะได้เป็นกระบวนการหมายเลข 4.1.1, 4.1.2, 4.1.3 เป็นต้น กระบวนการ 4.1.1 เมื่อนำมาแยกต่อจะได้ 4.1.1.1, 4.1.1.2, 4.1.1.3 เป็นต้น การแยกองค์ประกอบของแผนภาพการไหลของข้อมูลทำได้ไปเรื่อยๆ จนกว่าจะพอใจ หรือเมื่อได้รายละเอียดเพียงพอกับการนำไปออกแบบระบบ แผนภาพการไหลของข้อมูลที่จะเลียดมากจะช่วยให้การออกแบบระบบทำได้ดีขึ้น แต่จะเสียเวลาในการเขียนแผนภาพการไหลของข้อมูลมากขึ้นด้วย

2.2.3.1 แผนภาพการไหลของข้อมูลระดับที่ 1 (DFD Level 1)

แผนภาพการไหลของข้อมูลระดับที่ 1 จะแยกมาจากระดับที่ 0 จากตัวอย่างด้านล่างกระบวนการ 4.1, 4.2 และ 4.3 อยู่ในระดับที่ 1 ซึ่งแยกมาจากกระบวนการ 4.0 ซึ่งเป็นระดับที่ 0



รูปที่ 2.4 แผนภาพการไหลของข้อมูลระดับที่ 1 [1]

รูปที่ 2.4 เป็นการอธิบายรายละเอียดของกระบวนการที่ 4.0 ในรูปที่ 2.3 ซึ่งเป็น DFD ระดับที่ 0 โดยเริ่มต้นที่กระบวนการ 4.1 เรียกข้อมูลการขายและสินค้าคงคลัง ซึ่งเป็นการเรียกข้อมูลจำนวนสินค้าที่ขายรายวันจากแฟ้มข้อมูลการขายสินค้า และแฟ้มข้อมูลสินค้าคงคลัง จากนั้นจะส่งข้อมูลทั้งสอง (ข้อมูลสินค้าคงคลังและข้อมูลการขาย) ไปยังกระบวนการที่ 4.2 คือ สรุปยอดสินค้าที่ขายและคงเหลือ เพื่อรวบรวมข้อมูลทั้งหมดและทำการสรุปเพื่อส่งข้อมูลสรุปไปให้กระบวนการที่ 4.3 คือ เตรียมออกรายงาน ที่ทำหน้าที่จัดเตรียมรายงานและส่งรายงานนี้ไปให้ผู้จัดการต่อไป

2.2.3.2 แผนภาพการไหลข้อมูลระดับที่ n (DFD Level n)

การแจกแจงรายละเอียดของ DFD ในระดับที่ 2 ในรูปที่ 2.5 เริ่มจากกระบวนการที่ 4.3 ใน DFD ระดับที่ 1 (รูปที่ 2.4) ซึ่งแสดงกระบวนการย่อย 2 กระบวนการคือ กระบวนการที่ 4.3.1 จัดรูปแบบรายงานซึ่งรับข้อมูลสรุป จากนั้นส่งข้อมูลรวมที่จัดรูปแบบแล้ว เช่น การสร้างตาราง การรวมค่าต่างๆ หรือการหาค่าเฉลี่ย เป็นต้น ไปยังกระบวนการที่ 4.3.2 เพื่อพิมพ์รายงาน แล้วส่งรายงานนี้แก่ผู้จัดการร้านต่อไป

ในกรณีที่ยังต้องการรายละเอียดเพิ่มเติมจาก DFD ระดับที่ 2 ก็สามารถทำได้โดยการแจกแจงรายละเอียดของกระบวนการ 4.3.1 และ 4.3.2 ไปเป็น DFD ระดับที่ 3 คือ 4.3.1.1, 4.3.1.2 หรือ 4.3.2.1, 4.3.2.2 เป็นต้น และทำเช่นนี้ไปเรื่อยๆ จนกว่าจะได้รายละเอียดที่เพียงพอในการนำไปออกแบบและพัฒนาซอฟต์แวร์ต่อไป

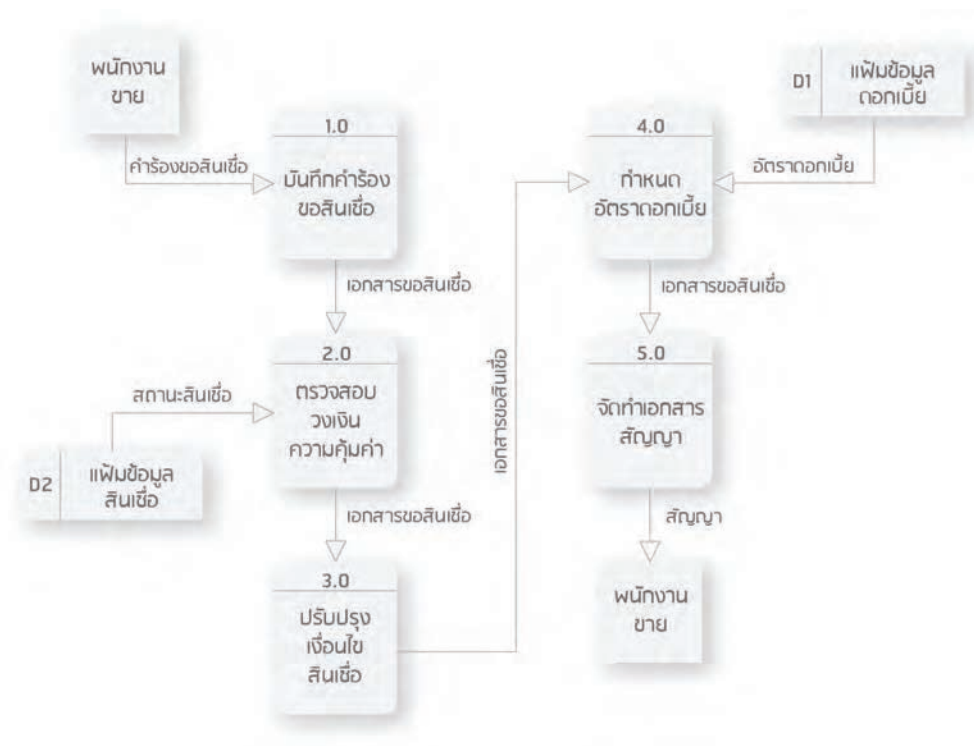
จากที่กล่าวมา จะสังเกตได้ว่าใน DFD แต่ละระดับนั้น ข้อมูลนำเข้า (Input) และส่งออก (Output) ของ DFD ในระดับที่ต่ำกว่าจะต้องสอดคล้องกับ DFD ที่ระดับสูงกว่าเสมอ ถ้าหากไม่สอดคล้องกันแสดงว่าการเขียน DFD มีความผิดพลาดและไม่ถูกต้อง



รูปที่ 2.5 แผนภาพการไหลของข้อมูลระดับที่ 2 [1]

2.3 ตัวอย่างการใช้แผนภาพการไหลของข้อมูลกับการปรับรื้อกระบวนการทางธุรกิจ (Using DFD in BPR)

แผนภาพการไหลของข้อมูลนอกจากจะใช้ในการวิเคราะห์สถานะของธุรกิจที่เป็นอยู่แล้วยังสามารถใช้เป็นเครื่องมือในการปรับรื้อกระบวนการทางธุรกิจ (Business Process Reengineering, BPR) หรือการปรับปรุงกระบวนการทางธุรกิจ (Business Process Improvement, BPI) [1] ได้โดยการวิเคราะห์และเขียน DFD ในสถานะเดิมก่อน จากนั้นก็ออกแบบกระบวนการใหม่ให้ลดขั้นตอนและมีประสิทธิภาพมากขึ้น ดังตัวอย่างด้านล่าง



รูปที่ 2.6 แผนภาพการไหลของข้อมูลก่อนการปรับรื้อกระบวนการทางธุรกิจ (BPR) [1]

จากรูปที่ 2.6 จะเห็นว่าในระบบเดิมนั้นมีทั้งหมด 5 กระบวนการ ซึ่งใช้เอกสารทั้งหมดไม่มีการใช้ระบบสารสนเทศเลย เริ่มต้นจากเมื่อพนักงานขายส่งคำขอสินเชื่อไปยังกระบวนการที่ 1.0 เพื่อบันทึกคำขอสินเชื่อ จากนั้นจัดทำเอกสารคำขอสินเชื่อ และส่งไปกระบวนการที่ 2.0 เพื่อตรวจสอบวงเงินและความคุ้มค่า โดยมีการเรียกดูข้อมูลจากแฟ้มข้อมูลสินเชื่อ เมื่อได้รับการอนุมัติก็จะส่งเอกสารดังกล่าวไปให้กระบวนการที่ 3.0 เพื่อปรับปรุงเงื่อนไขสินเชื่อ และส่งเอกสารไปยังกระบวนการที่ 4.0 เพื่อกำหนดอัตราดอกเบี้ยซึ่งจะใช้ข้อมูลจากแฟ้มข้อมูลดอกเบี้ย จากนั้นก็จะส่งไปยังกระบวนการที่ 5.0 เพื่อส่งเอกสารสัญญา และส่งให้กับพนักงานขายไปเสนอลูกค้าต่อไป

จากการวิเคราะห์ระบบเดิมพบว่า มีขั้นตอนมากและเสียเวลา ทำให้ลูกค้าไม่พอใจและสูญเสียโอกาสทางธุรกิจ องค์กรนี้จึงมีการปรับปรุงกระบวนการทำงานใหม่ในรูปที่ 2.7 โดยลดขั้นตอนเหลือเพียง 1 ขั้นตอน โดยการใช้ระบบเทคโนโลยีสารสนเทศซึ่งจะเก็บข้อมูลเครดิต ข้อมูลดอกเบี้ย และมีผู้เชี่ยวชาญช่วยให้คำปรึกษา ทำให้ลดขั้นตอนลดเวลาการทำงาน และทำให้ลูกค้ามีความพึงพอใจมากขึ้น



รูปที่ 2.7 แผนภาพการไหลของข้อมูลหลังการปรับปรุงกระบวนการทางธุรกิจ (BPR) [1]

แบบฝึกหัดท้ายบท

ให้ใช้กรณีตัวอย่างนี้ในการตอบคำถาม

ร้านค้าโทรศัพท์มือถือจะส่งซื้อโทรศัพท์มือถือมาจากบริษัท AIX โดยระบุยี่ห้อ และรุ่นที่ต้องการ เมื่อโทรศัพท์มาที่ร้านก็จะรับและตรวจสอบกับการสั่งซื้อ จากนั้นก็จะตรวจสอบโทรศัพท์ว่าถูกต้องและใช้งานได้ นอกจากโทรศัพท์แล้วทางร้านยังต้องส่ง SIM card (ราคา 200 บาท) ซึ่งมี 2 แบบคือ แบบรายเดือน และแบบเติมเงิน (มีแบบ 100, 500 และ 1,000 บาท) โดยที่ใน SIM card จะมีหมายเลขโทรศัพท์บรรจุอยู่เรียบร้อยแล้ว เมื่อลูกค้ามาซื้อสินค้าที่ร้านก็จะเลือกโทรศัพท์ตามรุ่นและยี่ห้อที่ต้องการ จากนั้นก็จะเลือกประเภทของ SIM card เลือกเบอร์โทร และเลือกโปรโมชั่น ซึ่งมีเฉพาะอยู่ 3 แบบ จากนั้นก็จะชำระเงินค่าโทรศัพท์ และ SIM card ในกรณีที่เติมเงิน ต้องจ่ายเงินค่าบัตรด้วย เมื่อได้รับใบเสร็จรับเงินแล้วก็สามารถใช้โทรศัพท์ได้ทันที

การชำระเงินมี 2 แบบคือ 1. ชำระด้วยเงินสด 2. ชำระด้วยบัตรเครดิต ในกรณีที่ชำระด้วยบัตรเครดิต ระบบจะต้องไปติดต่อกับผู้ให้บริการบัตร เช่น VISA, AMEX เป็นต้น เพื่อตรวจสอบวงเงินของบัตร ถ้าผ่านจึงจะออกใบเสร็จรับเงินให้ ถ้าไม่ผ่านจะยกเลิกการขายให้กับลูกค้ารายนั้น ณ สิ้นวัน จะต้องรายงานสรุปยอดเงินที่ได้รับจากการขายจำนวนโทรศัพท์มือถือที่ขายไป และที่ยังเหลืออยู่โดยจำแนกเป็นยี่ห้อและรุ่น

เมื่อครบกำหนด 1 เดือนนับจากที่ร้านค้าได้ซื้อโทรศัพท์มือถือมาขาย ทางบริษัท AIX จะส่งใบแจ้งหนี้มาที่ร้าน ซึ่งทางร้านจะต้องชำระหนี้เป็นเงินสดให้กับพนักงานเก็บเงินของทาง AIX เมื่อชำระแล้ว พนักงานเก็บเงินจะเขียนใบเสร็จรับเงินให้

1. ให้เขียน Context Diagram ของระบบนี้
2. ให้เขียน Data Flow Diagram ใน Level 0 ของระบบนี้
3. ให้นำ Process การชำระเงินในข้อ 2 มาเขียน Data Flow Diagram ใน Level 1
4. การวิเคราะห์และออกแบบเชิงโครงสร้างเหมาะกับงานประเภทใด

เอกสารอ้างอิง

1. ปานใจ ชารัตตวงศ์. การวิเคราะห์และออกแบบระบบเทคโนโลยีสารสนเทศในมุมมองด้านการบริหาร. กรุงเทพฯ: สันทวีการพิมพ์, 2554.
2. Hoffer, Jeffrey A., Joey F. George, and Joseph S. Valacich. Modern Systems Analysis and Design. 5th ed. NJ.: Pearson Prentice Hall, 2008.
3. Schach, Stephen R. Object-Oriented and Classical Software Engineering. 8th ed. NY. McGraw-Hill, 2011.
4. Sommerville, Ian. Software Engineering. 8th ed. U.S.A.: Pearson Prentice Hall, 2007.
5. Sommerville, Ian. Software Engineering. 9th ed. U.S.A.: Pearson Prentice Hall, 2011.
6. Sommerville, Ian. Software Engineering. 10th ed. U.S.A.: Pearson Prentice Hall, 2016.
7. Wikipedia (2020). Structured analysis and design technique. [Online]. Available https://en.wikipedia.org/wiki/Structured_analysis_and_design_technique [2020, December,1].