การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps

นายเมธัส คำจาด

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
แขนงวิชาเทคโนโลยีสารสนเทศและการสื่อสาร สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช
พ.ศ. 2563

การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps

นายเมธัส คำจาด

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
แขนงวิชาเทคโนโลยีสารสนเทศและการสื่อสาร สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช
พ.ศ. 2563

SLA Enhancement Using Automation Server Based on CI/CD Concept for DevOps

Mr.Meitat Khamchaat

A Thesis Submitted in Partial Fulfillment of the Requirements for

the Degree of Master of Science in Information and Communication Technology

School of Science and Technology

Sukhothai Thammathirat Open University

2020

หัวข้อวิทยานิพนธ์	การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps
ชื่อและนามสกุล	นายเมธัส คำจาด
แขนงวิชา	เทคโนโลยีสารสนเทศและการสื่อสาร
สาขาวิชา	วิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช
อาจารย์ที่ปรึกษา	1. ผู้ช่วยศาสตราจารย์ ดร.ขจิตพรรณ กฤตพลวิมาน 2. รองศาสตราจารย์ ดร.วรัญญา ปุณณวัฒน์

วิทยานิพนธ์นี้ ได้รับความเห็นชอบให้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรระดับปริญญาโท เมื่อ วันที่ 26 สิงหาคม 2564

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.วิภา เจริญภัณฑารักษ์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ขจิตพรรณ กฤตพลวิมาน)

..... กรรมการ
(รองศาสตราจารย์ ดร.วรัญญา ปุณณวัฒน์)

..... ประธานกรรมการบัณฑิตศึกษา
(รองศาสตราจารย์ ดร.เทพศักดิ์ บุณยรัตพันธุ์)

ชื่อวิทยานิพนธ์ การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps

ผู้วิจัย นายเมธัส คำจาด **รหัสนักศึกษา** 2619600238

ปริญญา วิทยาศาสตรมหาบัณฑิต (เทคโนโลยีสารสนเทศและการสื่อสาร) **อาจารย์ที่ปรึกษา** (1) ผู้ช่วยศาสตราจารย์ ดร.ขจิตพรรณ กฤตพลวิมาน (2) รองศาสตราจารย์ ดร.วรัญญา ปุณณวัฒน์ **ปีการศึกษา** 2563

บทคัดย่อ

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อ (1) พัฒนาออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิด CI/CD สำหรับ DevOps (2) วิเคราะห์ประสิทธิภาพออโตเมชันเซิร์ฟเวอร์ และ (3) ประเมินผลการใช้งานออโตเมชันเซิร์ฟเวอร์

วิธีการดำเนินการวิจัย ประกอบด้วย (1) การติดตั้งและพัฒนาชุดทำงานออโตเมชันเซิร์ฟเวอร์ด้วยเครื่องมือเจ็นกิ้นและคำสั่งในรูปแบบไปป์ไลน์ และ (2) การทดสอบ วิเคราะห์ประสิทธิภาพโดยเจ็นกิ้นปลั๊กอินเชิงเวลา และประเมินผลการใช้งาน โดยใช้สถิติค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐาน การวิจัยครั้งนี้วัดจากปริมาณงานจำนวน 86 งาน และประเมินประสิทธิภาพจากผู้ใช้งานจำนวน 61 คน

ผลการวิจัยพบว่า อัตราส่วนความสำเร็จตามการสนับสนุนข้อตกลงระดับบริการ สำหรับการใช้งานออโตเมชันเซิร์ฟเวอร์และการทำงานแบบแมนนวล คิดเป็นร้อยละ 100 และ 1.16 ตามลำดับ ขนาดของซอฟต์แวร์ที่ใช้ในการทดสอบ กำหนดตามทฤษฎีโคโคโมเป็นขนาดกลาง ใหญ่ และเล็ก ได้อัตราส่วนความสำเร็จตามข้อตกลงระดับการบริการเป็น ร้อยละ 52.33 41.86 และ 5.81 ตามลำดับ การประเมินผลการใช้งานออโตเมชันเซิร์ฟเวอร์ 3 ด้าน มีดังนี้ ด้านการใช้งานและประสิทธิภาพของออโตเมชันเซิร์ฟเวอร์อยู่ที่ระดับมาก ($\bar{X}$ = 4.08 SD = 0.72) ด้านการลดระยะเวลาและขั้นตอนการทำงานจากการใช้งานอยู่ที่ระดับมาก ($\bar{X}$ = 4.05 SD = 0.72) และด้านการทำงานตามเป้าหมายของข้อตกลงระดับการบริการอยู่ที่ระดับมาก ($\bar{X}$ = 3.94 SD = 0.69) ผลการประเมินการใช้งานโดยรวมอยู่ที่ระดับมาก ($\bar{X}$ = 4.02 SD = 0.71) เช่นกัน

คำสำคัญ ออโตเมชันเซิร์ฟเวอร์ กระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง การพัฒนาและปฏิบัติการ ข้อตกลงระดับการบริการ

Thesis title: SLA Enhancement Using Automation Server Based on CI/CD Concept for DevOps

Researcher: Mr.Meitat Khamchaat; **ID:** 2619600238;

Degree: Master of Science (Information and Communication Technology);

Thesis advisor: (1) Dr. Khajitpan Kritpolviman, Assistant Professor;

(2) Dr.Waranya Poonnawat, Associate Professor; **Academic year:** 2020

Abstract

The objectives of this research were (1) to develop an automation server based on CI/CD concept for DevOps procedure to SLA enhancement; (2) to analyze the automation server performance; and (3) to evaluate the automation server usage.

Research methods consisted of (1) the installation and development of the automation server using Jenkins tools and pipeline script; and (2) testing, analyzing performance by Jenkins Plugin with time processing, and evaluating the usage by average and standard deviation statistics. This research measured the workload of 86 jobs. The performance was evaluated from 61 users.

The results showed that the success ratio to support the service level agreement for the deployment of the automation server and the manual procedure was 100% and 1.16%, respectively. The software size defined based on the COCOMO theory as medium, large, and small with the percentage of success regarding SLA of 52.33, 41.86 and 5.81, respectively. The evaluation of the automation server usage was presented in 3 aspects as follows: (1) the automation server usage and performance was at a high level ($\bar{X}$ = 4.08 *SD* = 0.72), (2) the reduction of time consumption and working procedures was at a high level ($\bar{X}$ = 4.05 *SD* = 0.72), and (3) the operation with achieving SLA criteria was at a high level ($\bar{X}$ = 3.94 *SD* = 0.69). The overall evaluation was also at a high level ($\bar{X}$ = 4.02 *SD* = 0.71).

Keywords: Automation Server, Continuous Integration, Continuous Delivery and Deploy, DevOps, SLA

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี เนื่องจากได้รับความกรุณาอย่างสูงจากผู้ช่วยศาสตราจารย์ ดร. ขจิตพรรณ กฤตพลวิมาน อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก และ รองศาสตราจารย์ ดร. วรัญญา ปุณณวัฒน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ที่ให้ความรู้ การแนะนำ คำปรึกษาด้านแนวคิด แนวทางและหลักการในการจัดทำวิทยานิพนธ์ฉบับนี้ ทั้งยังเป็นส่วนหนึ่งที่ช่วยในการตรวจสอบ รวมถึงการแก้ไขความถูกต้องในการจัดทำวิทยานิพนธ์ ตลอดจนให้คำแนะนำทางวิชาการที่ดี ผู้วิจัยขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ อีกทั้งขอขอบพระคุณประธานกรรมการสอบวิทยานิพนธ์และกรรมการสอบวิทยานิพนธ์ทุกท่าน ที่ได้ให้ข้อเสนอแนะและการชี้นำเพื่อการปรับปรุงวิทยานิพนธ์ให้ดียิ่งขึ้น

ขอขอบพระคุณประชากรและกลุ่มตัวอย่างทุกท่านที่ให้ความร่วมมือเป็นอย่างดีในการให้ข้อมูลเพื่อสร้างฐานความรู้ การพัฒนาชุดทำงานออโตเมชันเซิร์ฟเวอร์ ข้อตกลงระดับบริการ รวมถึงการช่วยเหลือในการเก็บข้อมูลในส่วนต่างๆ เพื่อให้วิทยานิพนธ์ฉบับนี้สมบูรณ์และบรรลุเป้าหมายตามแนวคิดและสมมติฐานที่วางไว้ ทั้งนี้ผู้วิจัยหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์แก่ผู้ที่สนใจและศึกษา ค้นคว้า หรือถูกนำไปใช้ในการอ้างอิงในการวิจัยด้านการพัฒนาออโตเมชันเซิร์ฟเวอร์ หากแต่วิทยานิพนธ์ฉบับนี้มีข้อบกพร่องประการใด ผู้วิจัยต้องขออภัยและน้อมรับความผิดพลาดนี้ไว้ หากมีโอกาสครั้งต่อไปจะได้นำไปปรับปรุงแก้ไขให้ดียิ่งขึ้นในอนาคต

เมธัส คำจาด

พฤษภาคม 2564

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ง
บทคัดย่อภาษาอังกฤษ.....จ
กิตติกรรมประกาศ.....ฉ
สารบัญตาราง.....ฌ
สารบัญภาพ.....ญ
บทที่ 1 บทนำ.....1
ความเป็นมาและความสำคัญของปัญหา.....1
วัตถุประสงค์การวิจัย.....12
กรอบแนวคิดการวิจัย.....12
ขอบเขตการวิจัย.....14
นิยามศัพท์เฉพาะ.....14
ประโยชน์ที่คาดว่าจะได้รับ.....16
บทที่ 2 วรรณกรรมที่เกี่ยวข้อง.....17
ระบบอัตโนมัติและออโตเมชันเซิร์ฟเวอร์.....17
กระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง (CI/CD).....21
ขั้นตอนการพัฒนาและปฏิบัติการ (Development and Operations).....24
ข้อตกลงระดับการบริการ (Service Level Agreements).....28
การประเมินขนาดซอฟต์แวร์ด้วยเทคนิคมาตรฐานการประมาณโมเดล COCOMO.....33
งานวิจัยที่เกี่ยวข้อง.....36
บทที่ 3 วิธีดำเนินการวิจัย.....40
ประชากรและกลุ่มตัวอย่าง.....40
เครื่องมือการวิจัย.....41
วิธีการรวบรวมข้อมูล.....50
วิธีการวิเคราะห์และนำเสนอข้อมูล.....52

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการดำเนินการวิจัย 54
ผลการพัฒนาออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ 54
ผลการวิเคราะห์ประสิทธิภาพของออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ 60
ผลการประเมินผลการใช้งานของออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ 63
บทที่ 5 สรุปการวิจัย อภิปรายผล และข้อเสนอแนะ 74
สรุปผลการวิจัย 74
อภิปรายผลการวิจัย 76
ข้อจำกัดของการวิจัย 78
ข้อเสนอแนะ 78
บรรณานุกรม 80
ภาคผนวก 88
ภาคผนวก ก แบบสรุปผลการเก็บข้อมูลตามระเบียบวิธีวิจัยวิทยานิพนธ์ 89
ภาคผนวก ข แบบประเมินการใช้งาน Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน Service Level Agreements ทางธุรกิจ 94
ประวัติผู้วิจัย 98

สารบัญตาราง

หน้า

ตารางที่ 1.1 ตารางตัวอย่างของ Service Level Agreements (SLA) ด้านเทคโนโลยีสารสนเทศ.....4
ตารางที่ 1.2 ตารางสถิติการติดตั้งระบบเพื่อใช้งานจริง ของหน่วยงานขนาดใหญ่ในปี 2555.....6
ตารางที่ 2.1 ตัวอย่างข้อกำหนดระดับการบริการทางธุรกิจในระดับองค์กร 30
ตารางที่ 2.2 ตัวอย่างข้อกำหนดระดับการบริการทางธุรกิจในระดับบริการ (NOSTRA, 2563).....32
ตารางที่ 3.1 ตัวอย่างการแบ่งกลุ่มขนาดของซอฟต์แวร์ระบบ 41
ตารางที่ 4.1 ปริมาณงานตามขนาดของประเภทซอฟต์แวร์หรือแอปพลิเคชันด้วยการแมนนวล 61
ตารางที่ 4.2 ปริมาณงานตามขนาดของประเภทซอฟต์แวร์หรือแอปพลิเคชันด้วยออโตเมชันเซิร์ฟเวอร์ 62
ตารางที่ 4.3 ผลสรุปการประเมินประสิทธิภาพการใช้งานของออโตเมชันเซิร์ฟเวอร์.....63
ตารางที่ 4.4 แบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ ตอนที่ 2 69
ตารางที่ 4.5 ข้อคิดเห็นหรือหรือเสนอแนะ 70
ตารางที่ 4.6 ผลสรุปสถิติการติดตั้งระบบเพื่อใช้งานจริงของหน่วยงานในการใช้งานและทดสอบประสิทธิภาพด้วยออโตเมชันเซิร์ฟเวอร์ 71
ตารางที่ 4.7 แบบสรุปผลการเก็บข้อมูลการทดสอบประสิทธิภาพอย่างย่อจากภาคผนวก ก ลำดับที่ 21 ถึง 28 72

สารบัญภาพ

หน้า

ภาพที่ 1.1 ความแตกต่างของยุคดิจิทัล 1.0 ถึง 4.0 2
ภาพที่ 1.2 กระบวนการ DevOps 8
ภาพที่ 1.3 ขั้นตอนแนวคิด CI/CD 9
ภาพที่ 1.4 กรอบแนวคิดการวิจัย 13
ภาพที่ 2.1 PDF Automation Server 20
ภาพที่ 2.2 Data Center Automation 21
ภาพที่ 2.3 กระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่องบน Azure Server 24
ภาพที่ 2.4 แนวทางการประยุกต์ขั้นตอนการพัฒนาและปฏิบัติการ : DevOps 26
ภาพที่ 2.5 ตัวอย่างข้อกำหนดระดับการบริการทางธุรกิจในระดับลูกค้า 30
ภาพที่ 3.1 กระบวนการทำงานตามแนวคิดการวิจัย 44
ภาพที่ 3.2 ขั้นตอนการทดสอบและใช้งานชุดทำงานออโตเมชันเซิร์ฟเวอร์ตามแนวคิดการวิจัย 46
ภาพที่ 3.3 การแสดงระยะเวลาเฉลี่ยรวมแต่ละพื้นที่และระยะเวลาเฉลี่ยรวมทุกพื้นที่ทำงาน ด้วย Jenkin Pipeline Plugin 48
ภาพที่ 4.1 ตัวอย่างการสั่งการกระบวนการบริหารจัดการชุดคำสั่งด้วย GitLab โดยเครื่องมือ Git Bash 55
ภาพที่ 4.2 กระบวนการติดตั้ง Windows Server 2019 Datacenter Student Version 56
ภาพที่ 4.3 Jenkins Plugin Manager 57
ภาพที่ 4.4 ตัวอย่างคำสั่ง Pipeline Script 58
ภาพที่ 4.5 ตัวอย่างการทำงานร่วมกันระหว่าง Jenkins และ Source Control ผ่าน Webhook 59
ภาพที่ 4.6 ตัวอย่างการทำงานตามขั้นตอนใน Pipeline Script ด้วย Jenkins 59
ภาพที่ 4.7 อัตราส่วนผู้ตอบแบบประเมินการใช้งานตามช่วงอายุ 64
ภาพที่ 4.8 อัตราส่วนผู้ตอบแบบประเมินการใช้งานตามระดับการศึกษา 65
ภาพที่ 4.9 อัตราส่วนผู้ตอบแบบประเมินการใช้งานตามตำแหน่ง/สายงาน 65

บทที่ 1
บทนำ

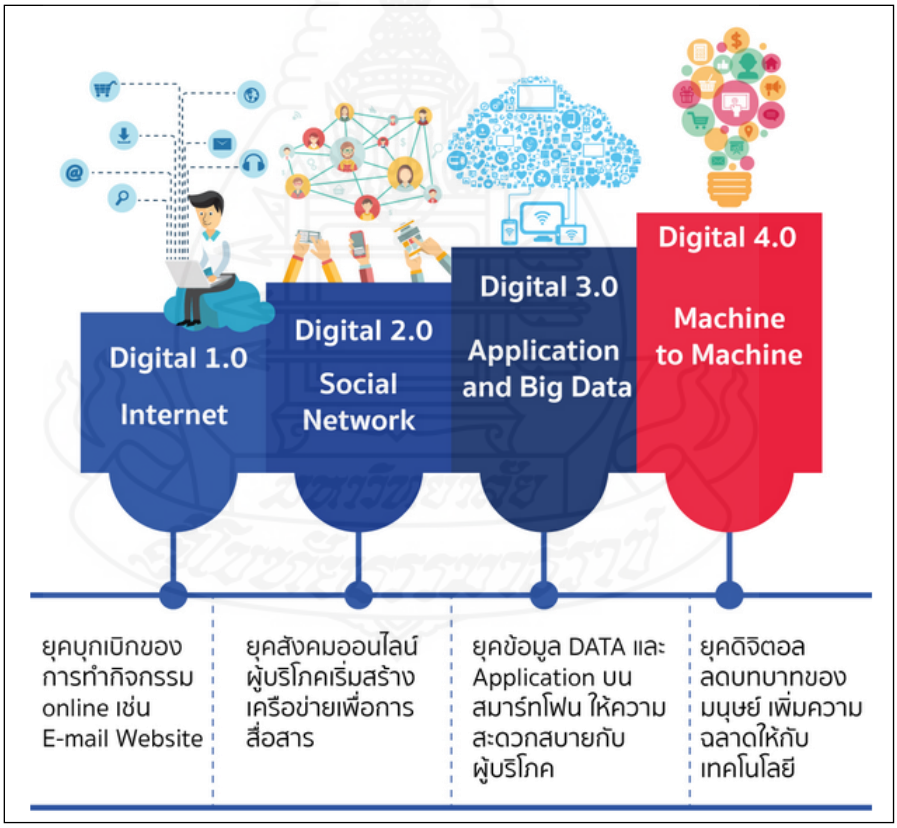
1. ความเป็นมาและความสำคัญของปัญหา

เทคโนโลยีที่พัฒนาขึ้นอย่างรวดเร็วเป็นส่วนหนึ่งของการพัฒนารูปแบบและมาตรฐานทางธุรกิจให้สอดคล้องกับความเปลี่ยนแปลงของโลกและวิวัฒนาการด้านต่างๆ ส่งผลให้ธุรกิจเกิดการปรับตัวก้าวเข้าสู่การพัฒนากระบวนการทางธุรกิจให้ผลักดันตนเองจากกระบวนการทางธุรกิจแบบเดิมเข้าสู่กระบวนการทางธุรกิจในโมเดลใหม่ ซึ่งมีกระบวนการธุรกิจอยู่มากในแต่ละช่วงเวลาที่เกิดการพัฒนาตามเทคโนโลยีเรียกว่า “ช่วงเวลาทางธุรกิจ” (DIP-SME ACAADEMY, 2561) เมื่อก้าวเข้าสู่การเปลี่ยนแปลงทางเทคโนโลยีการสร้างนวัตกรรมทางดิจิทัลจึงเกิดการแข่งขันขึ้นเพื่อความคล่องตัวที่จะช่วยให้กระบวนการทางธุรกิจเป็นธุรกิจรูปแบบใหม่เรียกว่า “ธุรกิจดิจิทัล” (Jorge Lopez, 2557)

1.1 การเข้าสู่ธุรกิจ Digital 4.0

จากราชกิจจานุเบกษา เรื่องนโยบายและแผนระดับชาติว่าด้วยการพัฒนาดิจิทัลเพื่อเศรษฐกิจและสังคมในส่วนที่ 3 ยุทธศาสตร์การพัฒนาดิจิทัลเพื่อเศรษฐกิจและสังคม ยุทธศาสตร์ที่ 2: ขับเคลื่อนเศรษฐกิจด้วยเทคโนโลยีดิจิทัล ซึ่งในแผนงานได้ระบุใจความสำคัญไว้ว่า “ 1) เพิ่มขีดความสามารถในการแข่งขันของภาคธุรกิจตลอดห่วงโซ่คุณค่า โดยผลักดันธุรกิจให้เข้าสู่ระบบการค้าดิจิทัลสู่สากล และให้เกิดการใช้เทคโนโลยีและข้อมูลเพื่อปฏิรูปการผลิตสินค้าและบริการ 2) เร่งสร้างธุรกิจเทคโนโลยีดิจิทัล (digital technology startup) ให้เป็นฟันเฟืองสำคัญในการขับเคลื่อนเศรษฐกิจดิจิทัล 3) พัฒนาอุตสาหกรรมเทคโนโลยีดิจิทัลของไทยให้มีความเข้มแข็งและสามารถแข่งขันเชิงนวัตกรรมได้ในอนาคต โดยเฉพาะอย่างยิ่งอุตสาหกรรมที่ไทยมีศักยภาพและเป็นอุตสาหกรรมแห่งอนาคต 4) เพิ่มโอกาสทางอาชีพเกษตรและการค้าขายสินค้าของชุมชนผ่านเทคโนโลยีดิจิทัล โดยดำเนินการร่วมกันระหว่างหน่วยงานจากทั้งภาครัฐ ภาคเอกชนและภาคประชาชน” (สำนักงานคณะกรรมการดิจิทัลเพื่อเศรษฐกิจและสังคมแห่งชาติ, 2561) เห็นได้อย่างชัดเจนว่ารัฐบาลได้ให้ความสำคัญต่อการเปลี่ยนแปลงของเทคโนโลยีและโลกให้ทันตามยุคสมัยปัจจุบัน ธุรกิจด้านต่างๆ จึง

มุ่งเน้นที่จะพัฒนาธุรกิจของตนเองให้ก้าวทันตามนโยบาย Thailand 4.0 เพื่อมุ่งหวังให้ธุรกิจของตนเองเกิดการใช้เทคโนโลยีในการเพิ่มขีดความสามารถของธุรกิจผ่านเทคโนโลยีดิจิทัลหรือ “ธุรกิจ Digital 4.0” อันเป็นฟันเฟืองสำคัญในการพัฒนาประเทศตามนโยบายการขับเคลื่อนเศรษฐกิจด้วยเทคโนโลยีดิจิทัล เมื่อต้องการให้ธุรกิจสามารถพัฒนาเข้าสู่ Digital 4.0 ได้ การศึกษาประเด็นสำคัญของการพัฒนาธุรกิจสู่ Digital 4.0 จึงเป็นส่วนหนึ่งที่ธุรกิจต้องทำความเข้าใจ เริ่มจาก Digital 1.0 อันเป็นยุคบุกเบิกสู่การทำกิจกรรมทางด้านออนไลน์ ซึ่งอาศัยอินเทอร์เน็ตในการดำเนินธุรกิจ, Digital 2.0 เป็นยุคที่ธุรกิจต้องปรับตัวเข้าสู่สังคมออนไลน์มุ่งสู่เครือข่ายทางการสื่อสาร, Digital 3.0 เป็นยุคที่ต้องอาศัยข้อมูลและแอปพลิเคชันเพื่ออำนวยความสะดวกแก่ผู้บริโภค และยุค Digital 4.0 ซึ่งเป็นยุคเป้าหมายในหัวข้อการวิจัยและจัดทำวิทยานิพนธ์ เป็นยุคที่มุ่งหวังให้ลดบทบาทของมนุษย์ โดยใช้เทคโนโลยีเข้ามาแทนที่ในการลดขั้นตอน เวลา รวมถึงต้นทุนค่าใช้จ่ายและประสิทธิภาพในการทำงาน ดังภาพที่ 1.1



ภาพที่ 1.1 ความแตกต่างของยุคดิจิทัล 1.0 ถึง 4.0
(สำนักงานปลัดกระทรวงศึกษาธิการ, 2561)

1.2 ความสำคัญของข้อตกลงระดับบริการทางธุรกิจ (Service Levels Agreement : SLA)

เมื่อธุรกิจจำเป็นต้องก้าวข้ามกระบวนการธุรกิจเดิมเพื่อการปรับตัวเข้าสู่ยุค Digital 4.0 ธุรกิจจำเป็นต้องอาศัยความฉลาดของเทคโนโลยีให้สามารถดำเนินการในกระบวนการต่างๆ ได้อย่างอัตโนมัติ ซึ่งซอฟต์แวร์ถือเป็นสิ่งสำคัญสิ่งหนึ่งที่จำเป็นต้องใช้ในการบริหารจัดการทางธุรกิจ สำหรับเพิ่มศักยภาพการแข่งขันให้ทันต่อความต้องการของตลาดในด้านการตอบสนองลูกค้ารายบุคคล, ช่องทางการติดต่อสื่อสาร รวมถึงการบูรณาการการใช้ทรัพยากรองค์รวมให้เกิดประโยชน์สูงสุดโดยใช้ซอฟต์แวร์เป็นส่วนควบคุมผ่านเครือข่ายอินเทอร์เน็ต (บริษัท สุมิพล คอร์ปอเรชั่น จำกัด, 2019) และด้วยโครงสร้างองค์กรธุรกิจด้านเทคโนโลยีสารสนเทศที่ถือเป็นองค์ประกอบสำคัญในการนำเอาเทคโนโลยีสารสนเทศมาปรับใช้กับองค์กรธุรกิจ Digital 4.0 อันทำให้ธุรกิจสามารถขับเคลื่อนไปได้ตามทิศทางที่กำหนดได้อย่างมีประสิทธิภาพ เป็นผลดีแก่ผู้รับบริการและผู้มีส่วนได้เสียในธุรกิจ การเริ่มต้นปรับกระบวนการอัตโนมัติสู่ธุรกิจ Digital 4.0 จึงควรเริ่มจากทางด้านเทคโนโลยีสารสนเทศภายในธุรกิจก่อนและคำนึงถึงผู้บริโภคเป็นหลัก (ธนาวิชญ์ จินดาประดิษฐ์, 2561) กระบวนการด้านเทคโนโลยีสารสนเทศโดยเฉพาะอย่างยิ่งด้านซอฟต์แวร์นั้นถือเป็นช่องทางในการประสานงานและดำเนินการระหว่างผู้บริโภคหรือผู้รับบริการในปัจจุบัน ซึ่งสามารถใช้เป็นช่องทางการตลาดให้แก่ธุรกิจได้อีกด้วย การดำเนินการทางธุรกิจโดยเฉพาะนโยบายทางด้านเทคโนโลยีสารสนเทศจึงต้องอาศัยการกำหนดข้อตกลงในการให้บริการ เพื่อการรักษาระดับคุณภาพการให้บริการทางด้านเทคโนโลยีสารสนเทศตามข้อตกลง หรือ Service Level Agreements (SLA) ในการควบคุมการทำงานให้เป็นไปตามข้อตกลงและเวลาที่กำหนด ทั้งนี้สำหรับการประเมินความมีประสิทธิภาพของแผนงานทางด้านเทคโนโลยีสารสนเทศ และเพื่อใช้เป็นแนวทางในการตัดสินใจในอนาคต รวมถึงการตั้งเป้าหมายใช้เป็นแนวทางให้การปฏิบัติการอย่างเหมาะสมหรือกล่าวได้ว่าเป็นกระบวนการที่ช่วยให้มั่นใจว่างานหรือบริการทั้งหมด สามารถให้บริการตามข้อตกลง ระหว่างผู้ให้บริการกับทางผู้รับบริการ ซึ่งจะต้องทำข้อตกลงกันเกี่ยวกับการใช้บริการในแต่ละบริการ โดยข้อตกลงระดับบริการของแต่ละหน่วยงานหรือแต่ละองค์กรอาจแตกต่างกันออกไปขึ้นอยู่กับยุทธศาสตร์ แผนงาน และนโยบายขององค์กรและหน่วยงานนั้นๆ ข้อตกลงระดับบริการมักจะถูกผูกกับตัวชี้วัดเพื่อนำไปใช้ในการประเมินประสิทธิภาพการทำงานขององค์กร หน่วยงาน รวมถึงบุคคล (องอาจ วรฤทธิ์โสภณ, 2559) ซึ่งมักรู้จักกันในชื่อ Key Performance Indicator หรือ KPI ดัชนีที่ใช้ชี้วัดผลงานหรือความสำเร็จของงาน โดยจะแสดงให้เห็นรายละเอียดในความสำเร็จหรือล้มเหลวของงาน KPI เป็นเทคนิควิธีการหนึ่งที่นิยมนำมาใช้ในการประเมินผลการปฏิบัติงานในปัจจุบัน โดยอาจวัดเป็นตัวบุคคลหรือหน่วยงาน ประกอบด้วย Key คือจุด

หลัก หัวข้อหลัก หรือเป้าหมายหลักหน่วยงาน, Performance คือประสิทธิภาพ ประสิทธิผล หรือผลของการกระทำ ความสามารถของบุคคลภายในหน่วยงาน และ Indicator คือตัวชี้วัดหรือดัชนีชี้วัด สามารถใช้วัดเวลาในการทำงานหรือผลงาน (MindPHP.COM, 2560) ดังนั้นหาก KPI ไม่ประสบความสำเร็จนั้นส่วนหนึ่งมีสาเหตุมาจากข้อตกลงระดับบริการและกระบวนการทางธุรกิจขององค์กรหรือหน่วยงานด้วย สำหรับตัวอย่างการกำหนดข้อตกลงระดับบริการที่สอดคล้องกับ KPI ตารางที่ 1.1

ตารางที่ 1.1 ตารางตัวอย่างของ Service Level Agreements (SLA) ด้านเทคโนโลยีสารสนเทศ

Business Unit : แผนกสารสนเทศ	
Service Statement Title : Application ERP Service	**Contact Point** : เจ้าหน้าที่แผนกสารสนเทศ Tel. 02-730-1240 ต่อ 416 Email : info@thaipolimer.co.th
Service Target : ให้บริการเกี่ยวกับโปรแกรมที่ใช้งานทางด้าน บัญชี จัดซื้อ ฝ่ายขาย	
Service Description :	▪ แก้ไขปัญหาต่างๆที่เกิดขึ้นกับระบบงานแอปพลิเคชัน ▪ ตรวจสอบปัญหาของแอปพลิเคชัน ▪ ให้การสนับสนุนการใช้งานแอปพลิเคชัน
Service : ผู้ขอใช้บริการ	
Service Hour : จันทร์ – ศุกร์ 8:00 น. – 17:00 น.	
KPI : สามารถแก้ไขปัญหาได้ในเวลาที่กำหนดตามระดับการให้บริการ	
Remark : การแก้ปัญหาในกรณีที่ไม่มีความรู้ในการแก้ปัญหาที่เกิดขึ้น ให้ดำเนินการส่งต่อปัญหาไปยังผู้ให้บริการภายนอก ที่คอยสนับสนุนการแก้ปัญหาโดยเร็วที่สุด	
Exception to erms and Conditions of SLA : ▪ กรณี มีปัญหาไม่สามารถใช้งานได้ทั้งองค์กร Critical TTR < 2 ชั่วโมง ▪ กรณี มีปัญหาไม่สามารถใช้งานได้บางส่วน High TTR < 4 ชั่วโมง ▪ กรณี มีปัญหาไม่สามารถใช้งานได้รายบุคคล TTR < 48 ชั่วโมง	

(องอาจ วรฤทธิ์โสภณ, 2559)

1.3 แนวคิดกระบวนการบูรณาการ กระบวนการการส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ

ปัญหาหลักของการทำให้ข้อตกลงระดับบริการทางด้านเทคโนโลยีสารสนเทศซอฟต์แวร์ไม่สำเร็จตามข้อกำหนดที่ได้กำหนดไว้นั้นส่วนใหญ่มักเกิดจากขั้นตอนและกระบวนการทำงานทางด้านเทคโนโลยี ซึ่งสอดคล้องกับข้อเสนอแนะของการวิจัยเรื่อง การพัฒนาระบบ IT Support ด้วยหลักการลีนเพื่อเพิ่มประสิทธิภาพการบริการกรณีศึกษาบริษัท D Thailand จำกัด โดย ธรรมรัตน์ ปัญญาภรณ์ประสาท และวรินทร์วงษ์มณี (2559) ที่กล่าวว่า “จุดด้อยของการที่ทำให้งานไม่บรรลุเป้าหมายตามกำหนดเวลาเกิดจากขั้นตอนการทำงาน” ทั้งยังมีงานวิจัยโดย องอาจ วรฤทธิ์โสภณ (2559) เรื่องการจัดกระบวนการให้บริการด้าน IT ให้มีมาตรฐานการทำงานโดยใช้หลักการ ITIL V.3 (กรณีศึกษา: บริษัท ไทยโพลิเมอร์ ซัพพลาย จำกัด) ที่ระบุว่า “การเกิดอุปสรรคความล่าช้าจากขั้นตอนการดำเนินงาน” รวมถึงข้อมูลตามทฤษฎีข้อจำกัด (Theory of Constraint) ที่มีการเขียนเป็นบทความไว้บนสื่อออนไลน์โดย Pariwat Saknimitwong (2561) กล่าวว่า “กระบวนการทำงานต่างๆ เช่น กระบวนการผลิต ทุกๆกระบวนการจะมีคอขวดอยู่อย่างน้อย 1 จุดเสมอ การปรับปรุงอัตราเร็วของกระบวนการจะเกิดขึ้นได้ก็ต่อเมื่อคอขวดถูกปรับปรุง การเพิ่มอัตราเร็วในส่วนอื่นที่ไม่ใช่คอขวดของกระบวนการจะไม่มีผลต่ออัตราเร็วของทั้งระบบ” และ April, A. และคณะ (2543) กล่าวว่า “กระบวนการทำงานแบบส่งต่องานมักส่งผลต่อคุณภาพและคุณค่าทางธุรกิจ จากองค์ประกอบที่เกี่ยวข้องในการทำงานจำนวนมากจะทำให้ประสิทธิภาพในการแก้ปัญหาบรรลุเป้าหมายได้น้อยหรือล้มเหลว” ทำให้เห็นว่ากระบวนการที่เกิดขึ้นในทางเทคโนโลยีสารสนเทศด้านซอฟต์แวร์นั้นเป็นส่วนหนึ่งของกระบวนการผลิตและเป็นจุดที่ทำให้เกิดปัญหาความล่าช้า และเป็นปัญหาส่วนใหญ่ที่ทำให้ซอฟต์แวร์ที่พัฒนาหรือถูกแก้ไขใช้เวลานาน (Saknimitwong, P., 2561) ตัวอย่างเช่น ขั้นตอนจัดการปัญหาของซอฟต์แวร์ที่พบเมื่อได้รับแจ้งปัญหาโดย เมื่อพบปัญหาจากการใช้งานหรือข้อผิดพลาดภายในซอฟต์แวร์ จำเป็นต้องหยุดหรือระงับการทำงานของซอฟต์แวร์ดังกล่าวไว้ เพื่อไม่ให้ข้อมูลที่เกิดขึ้นมีความผิดพลาดน้อยที่สุด และเป็นไปตามข้อตกลงระดับบริการซึ่งต้องทำการแก้ไขและกลับมาใช้งานได้ภายใน 2 ชั่วโมง แต่กระบวนการในการดำเนินการเริ่มจาก 1) การหาข้อผิดพลาดของปัญหา 2) การแก้ไขข้อผิดพลาด 3) การ Scan Source Code ที่แก้ไข เพื่อหาช่องโหว่ของชุดคำสั่ง 4) การ Build Source Code เพื่อนำไปใช้งาน 5) การทำการทดสอบย่อยหรือ Unit Test 6) การทำการทดสอบเพื่อยอมรับการใช้ระบบจากผู้ใช้งานหรือ User Acceptance Test 7) ประสานงานทีมปฏิบัติการเพื่อจัดวางซอฟต์แวร์ในการนำไปใช้งานจริงบนเครื่องเซิร์ฟเวอร์สำหรับเผยแพร่ให้นำไปใช้งาน ด้วยเหตุผลด้าน

ความปลอดภัย ซึ่งไม่สามารถดำเนินการจากระบบเครือข่ายภายนอกบริษัทได้ 8) ทำการจัดวางและการเผยแพร่ระบบเพื่อใช้งานจริง เป็นต้น สามารถเห็นได้จากสถิติการปล่อยผลิตภัณฑ์ซอฟต์แวร์ที่มีการพัฒนาหรือแก้ไข เพิ่มหรือลดคุณสมบัติของซอฟต์แวร์ ดังตารางที่ 1.2

ตารางที่ 1.2 ตารางสถิติการติดตั้งระบบเพื่อใช้งานจริง ของหน่วยงานขนาดใหญ่ในปี 2555

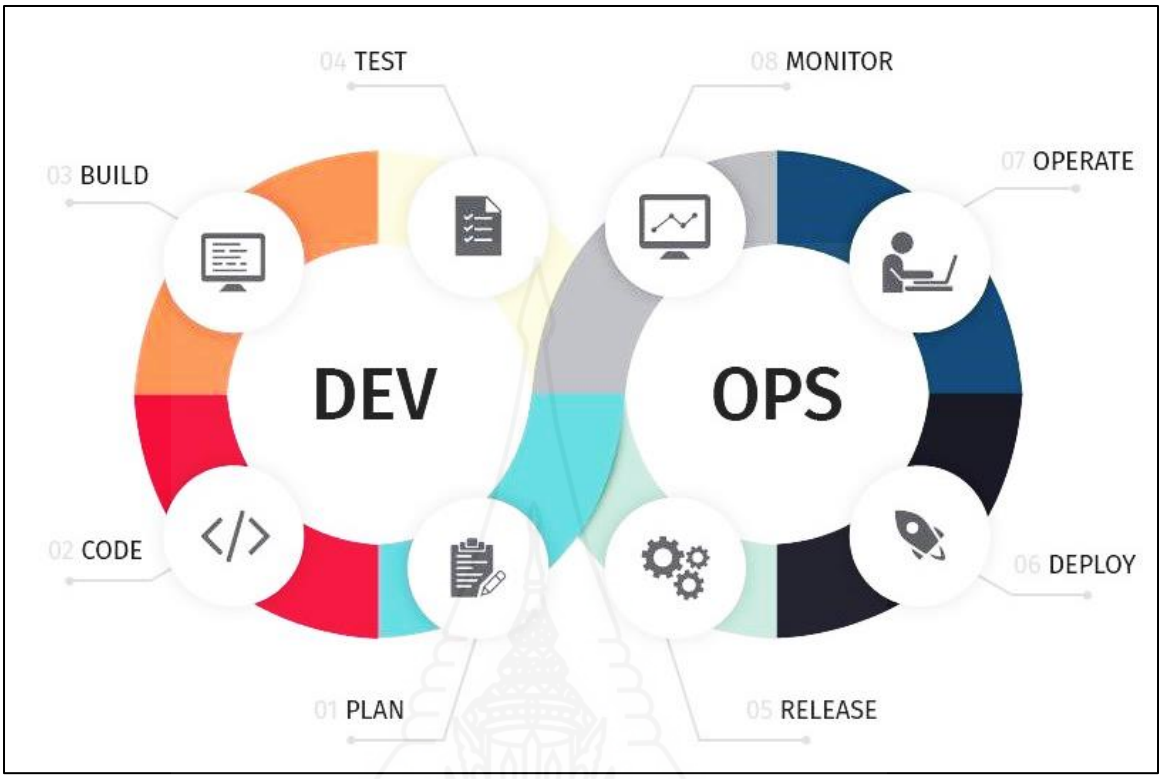
Company	**Deploy Frequency**	**Deploy Lead Time**	**Reliability**	**Customer Responsiveness**
Amazon	23,000 / Day	Minutes	High	High
Google	5,500 / Day	Minutes	High	High
Netflix	500 / Day	Minutes	High	High
Facebook	1 / Day	Hours	High	High
Twitter	3 / Week	Hours	High	High
Typical Enterprise	Once every 9 months	Month or Quarters	Low/Med	Low/Med

(Saknimitwong, P., 2561)

จากตารางที่ 1.2 แสดงให้เห็นถึงการเปลี่ยนแปลงและความถี่ในการจัดวางและเผยแพร่การแก้ไขหรือเวอร์ชันการทำงานใหม่ของซอฟต์แวร์ที่มีจำนวนมาก ซึ่งการใช้งานหรือการแก้ไขส่วนใหญ่มีผลกระทบต่อการใช้งานของลูกค้าผู้ใช้งาน หากไม่ทำการแก้ไขในทันทีจะทำให้ส่งผลกระทบในการใช้งานต่อผู้ใช้งานอันนำไปสู่ความผิดพลาดที่ต่อเนื่องกันได้ อาทิ ข้อมูลที่ถูกจัดเก็บลงในระบบที่ไม่ถูกต้องจากผลของการทำงานที่ผิดพลาด การแสดงผลค่าที่เกิดขึ้นจากการแสดงผลที่ผิดพลาด หรือรวมไปถึงการนำค่าที่เกิดจากการทำงานที่ผิดพลาดไปประมวลผลต่อที่ส่งผลให้เกิดเป็นความผิดพลาดที่จำเป็นต้องแก้ไขภายหลังอีกจำนวนมาก หากไม่ได้รับการแก้ไขในทันที เป็นต้น จึงทำให้เกิดข้อตกลงระดับบริการขึ้นเพื่อให้เกิดการแก้ไขได้ทันท่วงที แต่ด้วยสาเหตุต่างๆ ในด้านเวลา ทีมงานหรือบริษทการทำงาน มักเป็นเหตุผลที่ทำให้ข้อตกลงระดับบริการที่ได้กำหนดไว้ไม่เป็นไปตามที่คาดหวัง

The Chartered Institute of Purchasing & Supply (2552) ได้กล่าวถึง สาเหตุที่ทำให้ข้อตกลงระดับบริการไม่ประสบความสำเร็จว่า สาเหตุของความล้มเหลวข้อตกลงระดับบริการประกอบด้วย ขาดความมุ่งมั่นจากลูกค้าและผู้ให้บริการ, โครงสร้างการสนับสนุนที่ไม่เพียงพอสำหรับการสอบความล้มเหลวในการใช้แนวคิดข้อตกลงระดับบริการผ่านทีมงานโครงการแต่งตั้งผู้จัดการข้อตกลงระดับบริการและจัดการประชุมทบทวนระดับบริการปกติเวิร์กโหลดข้อตกลงระดับบริการจำเป็นต้องมีระบบการรายงานเพิ่มเติมและภายในกำหนดราคาโอน ควรให้ความสนใจกับการชดเชยงานพิเศษดังกล่าวโดยการบรรเทาพนักงานที่เกี่ยวข้องกับหน้าที่ที่มีอยู่บางส่วน,ข้อตกลงระดับบริการมีรายละเอียดมากเกินไป และข้อตกลงระดับบริการนั้นมีรายละเอียดไม่มากพอ ซึ่งเมื่อนำมาสรุปรวมกันจะเห็นไว้ว่า กระบวนการต่างๆทางเทคโนโลยีมักมีรายละเอียดในการทำงานที่มีรายละเอียดมาก มีขั้นตอนที่ซับซ้อนและคลุมเครือ ทำให้ระยะเวลาในการบรรลุเป้าหมายข้อตกลงระดับบริการสำเร็จลดน้อยลงส่งผลกระทบกับ KPI ไม่ผ่านตัวชี้วัดทั้งรายบุคคลและหน่วยงาน เพราะต้องดำเนินการทุกกระบวนการด้วยวิธีการแบบแมนวล (Manual) และด้วยระยะเวลาที่กระชั้นชิดอาจทำให้เกิดปัญหาข้อผิดพลาดที่เกิดจากมนุษย์ (Human Error) ได้อีกด้วย (Joana Lourenço 2557) แต่ในธุรกิจ Digital 4.0 ที่ต้องอาศัยเทคโนโลยีอัตโนมัติเข้ามาช่วยในการดำเนินการธุรกิจ มีโอกาสทำให้กระบวนการทำงานด้านเทคโนโลยีสารสนเทศของธุรกิจ Digital 4.0 ทำงานได้อย่างรวดเร็ว ทำให้ข้อตกลงระดับบริการสามารถบรรลุเป้าหมายและไม่ทำให้เกิดผลกระทบกับ KPI ของบุคคลหรือหน่วยงานได้ โดยอาศัยแนวคิด CI/CD ภายใต้กระบวนการของ DevOps (Saknimitwong, P., 2561) ในการพัฒนาออโตเมชันเซิร์ฟเวอร์ทำหน้าที่แทนทุกกระบวนการในขั้นตอนเดียว เพื่อให้ขั้นตอนและกระบวนการต่างๆที่เกิดขึ้นในทางเทคโนโลยีสารสนเทศลดลงทั้งในเชิงกระบวนการ รวมถึงระยะเวลาในการทำงานด้วย

กระบวนการทำงานตามขั้นตอนการพัฒนาและปฏิบัติการหรือ DevOps ถือเป็นรูปแบบ วิธีการปฏิบัติ วัฒนธรรม และกระบวนการต่างๆ เพื่อใช้แก้ไขปัญหาที่เกิดจากความขัดแย้งระหว่างขั้นตอนการพัฒนา (Development) และขั้นตอนการปฏิบัติการ (Operations) ช่วยเพิ่มประสิทธิภาพในการทำงานให้กับงานด้านเทคโนโลยีสารสนเทศออกสู่ตลาดได้รวดเร็วขึ้น มีคุณภาพและเสถียรภาพมากขึ้น เพื่อให้บริษัทสำเร็จตามเป้าหมายที่วางไว้

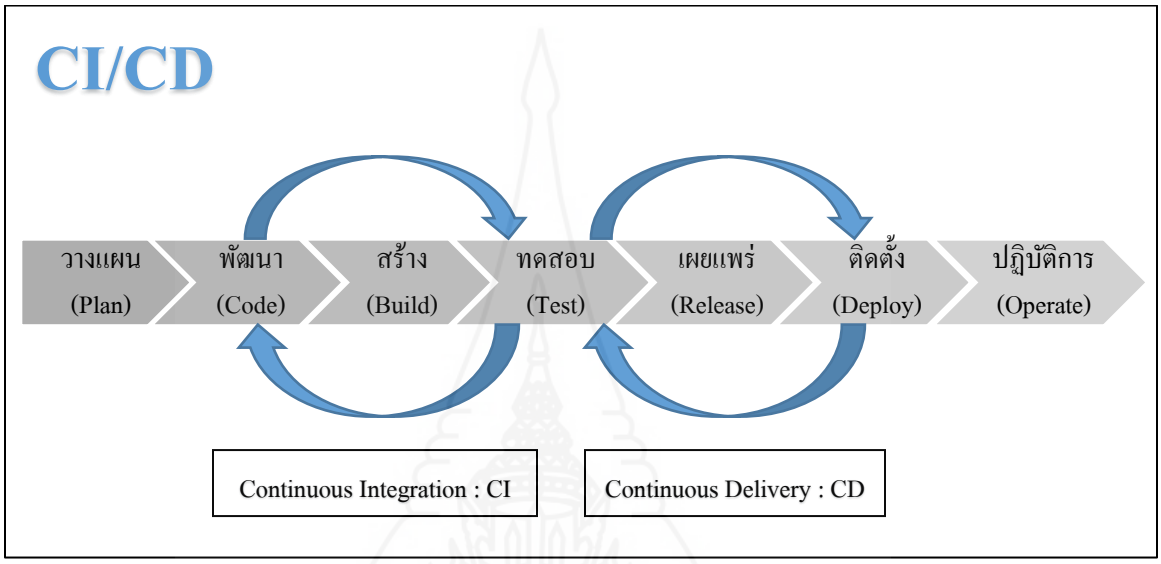


ภาพที่ 1.2 กระบวนการ DevOps

(Sakulsubwatthana, W., 2562)

จากภาพที่ 1.2 ได้แสดงให้เห็นถึงขั้นตอนต่างๆของ DevOps ซึ่งประกอบด้วย 1) การวางแผน (Plan) เป็นขั้นตอนการวางแผนโดยอาศัยเครื่องมือต่างๆในการวางแผน 2) การพัฒนา (Code) เป็นขั้นตอนของการพัฒนาชุดคำสั่งและบริหารจัดการชุดคำสั่งด้วยเครื่องมือเฉพาะทางที่เหมาะสม 3) การสร้างชุดทำงานโปรแกรม (Build) เป็นขั้นตอนของการประมวลผลและบีบอัดชุดคำสั่งเพื่อนำไปในกระบวนการเผยแพร่และติดตั้งใช้งานจริง 4) การทดสอบ (Test) เป็นขั้นตอนของการทดสอบการทำงานของชุดคำสั่ง 5) การเผยแพร่ (Release) เป็นขั้นตอนการติดตั้งระบบบนส่วนของเครื่องทรัพยากรตามสภาพแวดล้อมการทดสอบที่มีสภาพแวดล้อมการทำงานที่มีความใกล้เคียงกับเครื่องทรัพยากรตามสภาพแวดล้อมจริงที่มีการใช้งานหรือให้บริการอยู่ในปัจจุบันหรือถูกเรียกว่าเครื่อง Statging 6) การติดตั้ง (Deploy) เป็นขั้นตอนของการติดตั้งเพื่อนำไปใช้งาน (Deployment) สู่สภาพแวดล้อมการทำงานและใช้งานบนเครื่องที่ถูกใช้งานหรือให้บริการอยู่จริงหรือถูกเรียกว่าเครื่อง

Production 7) การปฏิบัติการ (Operate) เป็นขั้นตอนที่ช่วยให้ทีมปฏิบัติการ (Operation) สามารถบริหารจัดการกระบวนการและทำงานต่อได้อย่างเหมาะสม 8) การติดตาม (Monitor) เป็นขั้นตอนที่ใช้ในการสนับสนุนและติดตามของทีมปฏิบัติการในการติดตามความเสี่ยงหรือปัญหาที่เกิดขึ้นในระบบ



ภาพที่ 1.3 ขั้นตอนแนวคิด CI/CD

(Virga, M. and Clark, L., 2562)

จากภาพที่ 1.3 ได้แสดงให้เห็นถึงแนวคิดกระบวนการบูรณาการ (Continuous Integration Process) และ แนวคิดกระบวนการการส่งมอบและปรับใช้แบบต่อเนื่อง (Continuous Delivery and Deploy Process) หรือมักถูกเรียกว่า CI/CD จะถูกนำมาใช้ร่วมกับขั้นตอนการพัฒนาและปฏิบัติการ (DevOps) เพื่อสนับสนุนกระบวนการอัตโนมัติตามยุค Digital 4.0 สำหรับการพัฒนา (Plan), การสร้างชุดทำงานโปรแกรม (Build), การทดสอบ (Test), การเผยแพร่ (Release) และ การติดตั้ง (Deploy) ให้กระบวนการต่างๆสามารถทำงานได้อย่างต่อเนื่อง หลังการพัฒนาเทคโนโลยีสารสนเทศด้านซอฟต์แวร์ โดยไม่จำเป็นต้องอาศัยทีมปฏิบัติการในการช่วยทำงานในส่วนของการติดตั้ง (Deployment) ตามกระบวนการเทคโนโลยีแบบเดิมที่ทำงานลักษณะแบบแมนวล (Manual) สามารถนำไปประยุกต์ให้ลดขั้นตอนการทดสอบด้วยการพัฒนาการทดสอบบแบบอัตโนมัติ (Automation Testing) ผ่านเครื่องมือเฉพาะทางได้ ทำให้การทำงานรวดเร็วขึ้นภายในขั้นตอนเดียว แต่เนื่องจากทั้ง

แนวคิด CI/CD และแนวคิดกระบวนการ DevOps นั้นยังเป็นเพียงแนวคิดนิยามในการทำงานที่เป็นรูปแบบของโมเดลที่มีการผสมผสานระหว่างแนวคิด วัฒนธรรม การปฏิบัติ และเครื่องมือที่ช่วยเพิ่มความสามารถในการทำให้องค์กรสามารถส่งผลิตภัณฑ์ทางด้านซอฟต์แวร์ออกสู่ผู้บริโภคได้เร็วขึ้น

ซึ่งมาตรฐานและการทำงานทั้งหมดของกระบวนการตามแนวคิด CI/CD และ DevOps นั้น ยังต้องอาศัยมาตรฐานและความปลอดภัยตามวิธีการปฏิบัติของหน่วยงานนั้นๆ บนกระบวนการย่อยของ CI/CD และ DevOps เพื่อให้สามารถตรวจสอบและบังคับใช้การปฏิบัติตามกฎระเบียบแบบไดนามิกในหน่วยงานได้ด้วยโครงสร้างพื้นฐานของหน่วยงานนั้นเอง (Amazon Web Services, Inc., 2562) ซึ่งในแต่ละหน่วยงานมีพื้นฐาน โครงสร้าง และเครื่องมือในการพัฒนา รวมถึงฮาร์ดแวร์ที่มีความแตกต่างกัน กระบวนการตามแนวคิด CI/CD และ DevOps จึงมีความยืดหยุ่นในการนำไปประยุกต์ใช้ อาทิ การประยุกต์ใช้มาตรฐาน CMMI เข้ากับกระบวนการ DevOps ของหน่วยงานกลุ่มตัวอย่าง เนื่องจากในแต่ละกระบวนการย่อยของ DevOps เป็นกระบวนการทางด้านการพัฒนาวิศวกรรมซอฟต์แวร์ เป็นต้น

1.4 มาตรฐานกระบวนการพัฒนาซอฟต์แวร์

มาตรฐาน CMMI หรือ Capability Maturity Model Integration นั้นเป็นมาตรฐานที่เหมาะสมกับการพัฒนาซอฟต์แวร์ในทุกด้านทั้งด้านซอฟต์แวร์ ฮาร์ดแวร์ กระบวนการวิเคราะห์และออกแบบ รวมถึงกระบวนการรวมผลิตภัณฑ์แบบบูรณาการ จากระดับมาตรฐาน 5 ระดับ ได้แก่ ระดับที่หนึ่งระดับสมรรถนะ (Performed level) หรือระดับการปรับใช้ (Initial level) ซึ่งเป็นมาตรฐานระดับเริ่มต้น ระดับที่สองระดับการบริหารจัดการ (Managed level) เป็นมาตรฐานเกี่ยวกับกระบวนการกำหนดกระบวนการบริหารจัดการขั้นพื้นฐาน 7 กลุ่ม ตั้งแต่ขั้นตอนการเก็บความต้องการไปถึงการบำรุงรักษา ระดับที่สามระดับการกำหนดกระบวนการ (Defined Level) เป็นระดับที่มีการกำหนดกระบวนการในทุกองค์ประกอบของงาน ระดับที่สี่ระดับจัดการเชิงปริมาณ (Quantitatively Managed Level) เป็นระดับที่นำหลักการจัดการเชิงปริมาณเข้ามาช่วยในจัดการตรวจสอบหาสาเหตุความผิดปกติ และระดับที่ห้าระดับการแก้ปัญหา (Optimizing level) เป็นระดับมาตรฐานเพื่อการวิเคราะห์หาสาเหตุสำหรับนำไปใช้แก้ปัญหา (มหศักดิ์ เกตุฉ่ำ, 2562) หรือแม้แต่การประยุกต์ใช้มาตรฐาน ISO-27001:2013 หมวดความสัมพันธ์ ISMS หรือ Information Security Management System ให้กับความสัมพันธ์ด้านการกำหนดนโยบายข้อตกลงระดับบริการให้เป็นไปตามแนวทางการบริหารจัดการความมั่นคงปลอดภัยในการให้บริการด้านซอฟต์แวร์ให้กับหน่วยงาน (บริษัท แอดวานซ์

อินโฟร์ เซอร์วิส จำกัด (มหาชน), 2560) ซึ่งนอกเหนือจากมาตรฐานต่างๆตามที่ได้ยกตัวอย่างมาแล้วนั้น ยังมีมาตรฐานอื่นๆ ที่สามารถนำมาใช้ประยุกต์ใช้กับแนวคิด CI/CD และ DevOps เพื่อกำหนดมาตรฐานในกระบวนที่ยืดหยุ่นให้กับกระบวนการทำงานภายในหน่วยงานต่างๆได้อีก เช่น ISO 19770-1:2017 หรือ SAM ที่เป็นมาตรฐานสำหรับการบริหารจัดการสินทรัพย์ซอฟต์แวร์ ISO 19770-2 สร้างมาตรฐานข้อมูลสำหรับจัดการข้อมูลที่กำหนดอัตลักษณ์ของซอฟต์แวร์ (SWID Tags) ((BSA | The Software Alliance, 2562) รวมถึงมาตรฐานทางด้านระบบเครือข่าย TCP/IP หรือมาตรฐานการควบคุมทางด้านเทคโนโลยีภายในของหน่วยงานที่สามารถนำไปประยุกต์ใช้ได้อีก

ดังนั้นจากการศึกษาแนวคิดกระบวนการบูรณาการ (Continuous Integration Process) และแนวคิดกระบวนการการส่งมอบและปรับใช้แบบต่อเนื่อง (Continuous Delivery and Deploy Process) (CI/CD) รวมถึงขั้นตอนการพัฒนาและปฏิบัติการ (DevOps) ถือเป็นกรรมวิธีหนึ่งที่สามารถช่วยเสริมสร้างธุรกิจในรูปแบบเดิมให้พัฒนาเป็นธุรกิจในรูปแบบ Digital 4.0 ได้ภายใต้กระบวนการอัตโนมัติ ซึ่งผู้วิจัยเล็งเห็นถึงความสามารถของกระบวนการ แนวคิดและเครื่องมือที่ใช้ในการพัฒนาประกอบกับความต้องการแก้ปัญหาที่เกิดขึ้นจากความไม่ประสบความสำเร็จในเป้าหมายของข้อตกลงระดับบริการทางด้านเทคโนโลยี ที่ส่งผลต่อตัวชี้วัดรายบุคคลหรือหน่วยงานในประสิทธิภาพการทำงานตามดัชนีชี้วัดผลงานหรือความสำเร็จของงาน (Key Performance Indicator : KPI) จึงมีความต้องการที่จะศึกษาวิจัย การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps โดยการพัฒนาออโตเมชันเซิร์ฟเวอร์ตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ เพื่อสนับสนุนข้อตกลงระดับการบริการทางธุรกิจในรูปแบบ Digital 4.0 ที่นำประโยชน์ของกระบวนการ DevOps และแนวคิด CI/CD มาใช้ในการสนับสนุนข้อตกลงระดับบริการจากการพัฒนาชุดจำลองออโตเมชันเซิร์ฟเวอร์สำหรับงานเทคโนโลยีสารสนเทศด้านซอฟต์แวร์ด้วยเครื่องมือต่างๆ ในการพัฒนาให้ธุรกิจรูปแบบเดิมพร้อมรองรับสู่ธุรกิจ Digital 4.0 ได้ โดยมีการทดสอบการใช้งานและประสิทธิภาพของออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps และประเมินผลเกี่ยวกับการลดระยะเวลาและขั้นตอนการทำงานทางธุรกิจในรูปแบบ Digital 4.0 ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ให้สนับสนุนข้อตกลงระดับบริการทางธุรกิจในการวิจัยครั้งนี้

2. วัตถุประสงค์การวิจัย

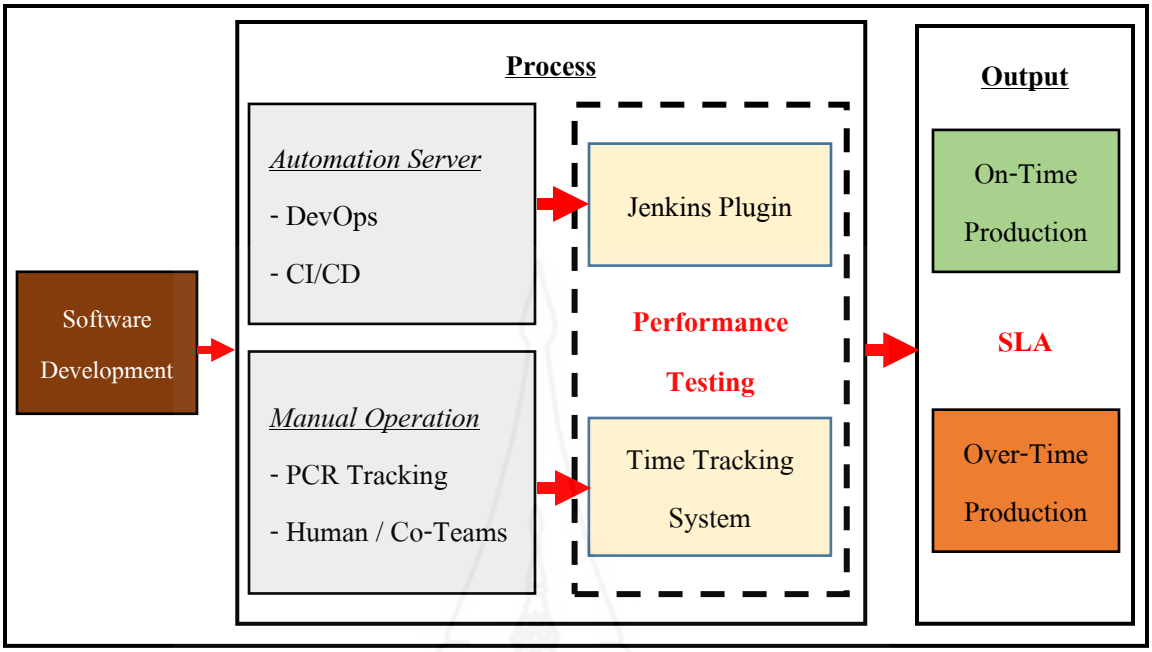
2.1 เพื่อพัฒนาออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ

2.2 เพื่อวิเคราะห์ประสิทธิภาพของออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ

2.3 เพื่อประเมินผลการใช้งานของออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ

3. กรอบแนวคิดการวิจัย

สำหรับการวิจัยเรื่องการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps ในครั้งนี้ เป็นการจัดทำการวิจัยเชิงทดลองเพื่อศึกษาถึงความสัมพันธ์ของกระบวนการทำงานในเชิงเปรียบเทียบระหว่างการทำงานรูปแบบอัตโนมัติ (Automation Server) และการทำงานรูปแบบแมนวล (Manual Step) ในด้านประสิทธิภาพของระยะเวลาในการสนับสนุนข้อตกลงระดับบริการหรือข้อตกลงระดับบริการด้วยการพัฒนาออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับกระบวนการ DevOps ซึ่งเน้นเป้าหมายในด้านระยะเวลาในการนำผลิตภัณฑ์ด้านซอฟต์แวร์ออกสู่ผู้ใช้งานเพื่อการใช้งานจริงภายใต้ระยะเวลาตามขอบเขตของข้อตกลงระดับบริการที่กำหนดโดยใช้เครื่องมือ Jenkins Plugin และระบบติดตามการทำงาน (Time Tracking) เป็นตัวชี้วัด ดังภาพที่ 1.4



ภาพที่ 1.4 กรอบแนวคิดการวิจัย

สำหรับกระบวนการทำงานตามกรอบแนวคิดการวิจัยที่จะเน้นการพัฒนาออโตเมชันเซิร์ฟเวอร์ให้ทำงานแบบอัตโนมัติเพื่อเพิ่มประสิทธิภาพข้อตกลงระดับการบริการ หลังจากที่ผู้พัฒนาดำเนินการพัฒนาหรือแก้ไขระบบงานแล้วเสร็จ ด้วยการเขียนคำสั่งการทำงานตามหลักการทำงานแบบ CI/CD ภายใต้กระบวนการทำงานตามแนวคิด DevOps อาศัยเครื่องมือ Jenkins ในการทำงานผ่านคำสั่งที่กำหนดขึ้นในรูปแบบไปป์ไลน์ (Pipeline) ในการติดต่อสื่อสารกันระหว่างเครื่องเซิร์ฟเวอร์ให้ทำงานต่อเนื่องตามกระบวนการตั้งแต่การสร้างชุดแพ็คเกจจากคำสั่งที่ผู้พัฒนาได้พัฒนาหรือแก้ไขขึ้นบนแหล่งจัดเก็บชุดคำสั่ง (Source Control) โดยอาศัยการทำงานการสร้างชุดแพ็คเกจด้วย MS-Build และส่งต่อไปยังกระบวนการจัดเก็บแยกเป็นเวอร์ชันก่อนส่งต่อไปยังเครื่องทำงานหลักเพื่อนำส่งผลิตภัณฑ์ที่มีการแก้ไขหรือพัฒนาขึ้นสู่เจ้าหน้าที่ตรวจสอบระบบและผู้ใช้งานต่อไป

4. ขอบเขตการวิจัย

การวิจัยครั้งนี้ผู้วิจัยได้ดำเนินการวิจัยโดยกำหนดกลุ่มตัวอย่างจากบุคลากรที่ประกอบอาชีพด้านเทคโนโลยีสารสนเทศ ที่มีส่วนเกี่ยวข้องกับการพัฒนาระบบ ทดสอบระบบ ระบบเครือข่าย ระบบปฏิบัติการ หรือติดตั้งระบบ เพื่อใช้งานในทางธุรกิจเป็นประชากรจำนวนทั้งสิ้น 86 คน เนื่องจากกลุ่มประชากรมีจำนวนไม่มากนักจึงนำมาใช้ในการกำหนดกลุ่มตัวอย่างทั้งหมด โดยกำหนดเป็นกลุ่มตัวอย่างด้วยจำนวนประชากรปริมาณงานทั้งหมด 86 งาน ซึ่งดำเนินการวิจัยโดยอาศัยเครื่องมือการวิจัย คือ 1) ชุดระบบทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ที่พัฒนาขึ้นด้วยคำสั่ง Groovy 2) โปรแกรมเสริมความสามารถ Jenkins Pipeline สำหรับพัฒนาชุดคำสั่งโปรแกรมเพื่อวัดประสิทธิภาพการทำงานของออโตเมชันเซิร์ฟเวอร์ 3) แบบบันทึกผลของกลุ่มตัวอย่างจากการทำงานด้วยวิธีการแบบแมนวล (Manual) และ 4) แบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุนข้อตกลงระดับบริการทางธุรกิจ ซึ่งการวิจัยในครั้งนี้เป็นการวิจัยเชิงทดลองโดยอาศัยการนำผลที่ได้จากเครื่องมือต่างๆ มาทำการวิเคราะห์และเปรียบเทียบประสิทธิภาพตามระยะเวลาที่มีความสอดคล้องกับความสัมพันธ์ในการสนับสนุนข้อตกลงระดับการบริการทางธุรกิจหรือ SLA ของแต่ละผลิตภัณฑ์ จากการคัดกรองกลุ่มตัวอย่างที่มีระยะเวลาการแก้ไขหรือพัฒนาระบบให้แล้วเสร็จตามข้อตกลงระดับบริการกรณีเร่งด่วนที่น้อยกว่าหรือเท่ากับ 3 ชั่วโมง ซึ่งการทำงานของออโตเมชันเซิร์ฟเวอร์ที่พัฒนาขึ้นนั้นจะทำงานอยู่บนพื้นฐานของระบบปฏิบัติการ Windows

5. นิยามศัพท์เฉพาะ

5.1 ออโตเมชันเซิร์ฟเวอร์ (Automation Server) หมายถึง กระบวนการทำงานที่พัฒนาขึ้นบนพื้นฐานการทำงานหลักของระบบปฏิบัติการ Windows Server สำหรับทำการประมวลผลชุดโปรแกรมที่นักพัฒนาพัฒนาขึ้นให้ใช้งานบนพื้นฐานการทำงานของระบบปฏิบัติการ Windows เป็นหลัก โดยจะทำงานแบบต่อเนื่องอัตโนมัติตามแนวคิด CI/CD เพื่อส่งมอบสู่ผู้ใช้งานตามเส้นทางของกระบวนการของหน่วยงานที่ทำการวิจัย ซึ่งเป็นการทำงานระหว่างเครื่องและเครื่อง (Machine to Machine) ด้วยการโปรแกรมคำสั่งร่วมกับเครื่องมือสำเร็จรูปหลังจากที่นักพัฒนาทำการเสร็จสิ้นการ

พัฒนาซอฟต์แวร์ต่างๆบนแหล่งจัดเก็บชุดคำสั่ง (Source Control) ที่กำหนดตามแนวคิดวิธีการการวิจัยในรูปแบบขั้นตอน DevOps

5.2 กระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง (Continuous Integration and Continuous Delivery and Deployment) หมายถึง กระบวนการซึ่งทำงานแบบอัตโนมัติต่อเนื่องภายใต้การโปรแกรมคำสั่งการทำงานด้วยภาษา Groovy ร่วมกับโปรแกรมร่วมทำงาน Jenkins ให้ทำงานตามแนวคิด CI/CD ตั้งแต่กระบวนการหลังจากนักพัฒนาเสร็จสิ้นการพัฒนาซอฟต์แวร์และส่งขึ้นบนแหล่งจัดเก็บชุดคำสั่ง (Source Control) ภายใต้ลำดับชั้นการจัดเก็บ (Repository) ที่กำหนด เพื่อประมวลผลการทำงานตามเส้นทางการส่งมอบซอฟต์แวร์สู่ผู้ใช้งานปลายทางตามแนวคิดขั้นตอน DevOps

5.3 ขั้นตอนการพัฒนาและปฏิบัติการ (Development and Operation : DevOps) หมายถึง แนวคิดกระบวนการ DevOps ซึ่งถูกออกแบบให้มีขั้นตอนการทำงานตามมาตรฐานการพัฒนาในแต่ละขั้นตอนของกระบวนการที่กำหนดขึ้นภายใต้โครงสร้างพื้นฐานของหน่วยงานที่ใช้ทำการวิจัย ถูกปรับการกระบวนการเส้นทางการทำงานตามเส้นทางการส่งมอบซอฟต์แวร์สู่ผู้ใช้งานปลายทางของหน่วยงาน

5.4 ข้อตกลงระดับการบริการ (Service Levels Agreement : SLA) หมายถึง ข้อตกลงระดับการบริการทางธุรกิจที่เกี่ยวข้องกับการใช้งานหรือแก้ไขทางเทคโนโลยีสารสนเทศด้านซอฟต์แวร์ที่มีความเร่งด่วนในการดำเนินการที่มีเป้าหมายในการแก้ไขและทำให้สามารถใช้งานได้หลังเกิดปัญหาหรือได้รับแจ้งปัญหาให้แล้วเสร็จภายในระยะเวลาไม่เกิน 3 วัน ซึ่งเป็นไปตามมาตรฐาน ISO-27001:2013 หมวดความสัมพันธ์ Information Security Management System (ISMS) ของหน่วยงานที่ใช้ทำการวิจัย

5.5 กระบวนการส่งมอบระบบแบบแมนวล (Manual Deployment) หมายถึง กระบวนการส่งมอบซอฟต์แวร์ด้วยการพัฒนาชุดคำสั่ง ผ่านการประมวลผลด้วยการใช้มนุษย์ทำงานในการสั่งการในทุกขั้นตอนวิธี ตามเส้นทางการส่งมอบซอฟต์แวร์สู่ผู้ใช้งาน ซึ่งถูกติดตามการทำงานด้วยระบบ PCR Tracking ตามเลขอ้างอิงการทำงานที่แสดงให้เห็นถึงผู้ปฏิบัติงาน และสถานะการทำงาน ตามเส้นทางการส่งมอบจนสิ้นสุดกระบวนการ

6. ประโยชน์ที่คาดว่าจะได้รับ

6.1 ได้ประสิทธิภาพที่เพิ่มขึ้นในการให้บริการงานด้านพัฒนาซอฟต์แวร์และเป็นไปตามข้อตกลงระดับบริการ จากการนำเอาออโตเมชันเซิร์ฟเวอร์มาปรับใช้ในกระบวนการทำงาน ทางเทคโนโลยี

6.2 ได้กระบวนการทำงานแบบอัตโนมัติในการทำงานได้อย่างต่อเนื่องตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการที่ช่วยลดและแก้ไขปัญหาจากการทำงานระหว่างทีมทำงานในแต่ละขั้นตอน อันส่งผลทำให้เกิดระยะเวลาในการรอคอยที่ส่งผลเสียต่อการนำซอฟต์แวร์ไปใช้งานจริงตามกำหนดเวลาตามข้อตกลงระดับการบริการ

6.3 ลดระยะเวลาในการทำงาน ระยะเวลาในการรอคอย รวมถึงระยะเวลาที่เกิดขึ้นจากการปฏิบัติงานของมนุษย์ที่ทำให้เกิดความล่าช้าด้วยขั้นตอนการทำงานแบบอัตโนมัติ เพื่อให้มนุษย์หรือทีมงานที่เกี่ยวข้องได้ทำหน้าที่ในการตรวจสอบและแก้ไขปัญหาจากการทำงานส่วนอื่นๆ ได้เพิ่มมากขึ้น

6.4 ได้รับผลการประเมินการทำงานภายในฝ่าย หน่วยงาน และทีมงานในระดับที่ดีขึ้น จากการทำงานด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้ต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ เพื่อสนับสนุนข้อตกลงระดับการบริการทางธุรกิจได้ตามระยะเวลาที่กำหนดไว้

บทที่ 2
วรรณกรรมที่เกี่ยวข้อง

การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps นี้ ได้มีการศึกษา ทบทวนทฤษฎีและวรรณกรรมต่างๆที่เกี่ยวข้อง เพื่อนำรายละเอียดและข้อมูลที่ได้ศึกษาและทบทวนวรรณกรรมมาใช้ประกอบการจัดทำการวิจัยโดยอ้างอิงตามหลักนิยาม แนวคิด ทฤษฎี และการปฏิบัติที่เกี่ยวข้อง ซึ่งสามารถแบ่งออกเป็นแนวคิด ทฤษฎี วรรณกรรม และงานวิจัยที่เกี่ยวข้องตามรายละเอียดต่อไปนี้

- ระบบอัตโนมัติและออโตเมชันเซิร์ฟเวอร์
- กระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง (CI/CD)
- ขั้นตอนการพัฒนาและปฏิบัติการ (Development and Operations)
- ข้อตกลงระดับการบริการทางธุรกิจ
- งานวิจัยที่เกี่ยวข้อง

1. ระบบอัตโนมัติและออโตเมชันเซิร์ฟเวอร์

1.1 ระบบอัตโนมัติ (Automation System)

ระบบอัตโนมัติ เป็นกระบวนการทำงานหนึ่งที่ทำงานด้วยพื้นฐานการทำงานตามบริบทของงาน ภายใต้การออกแบบการทำงานของระบบอัตโนมัติตามความต้องการและลักษณะการทำงานจากการทำงานด้วยมือหรือการทำงานเดิม โดยการปฏิบัติงานส่วนใหญ่เป็นการปฏิบัติงานจากการประมวลผลที่สั่งการด้วยระบบคอมพิวเตอร์ให้ทำงานอย่างต่อเนื่อง ซึ่งอาจมีมนุษย์เป็นเพียงผู้สั่งการเริ่มต้น เพื่อจุดประสงค์ในการลดกระบวนการทำงานของมนุษย์ สร้างความแม่นยำ ถูกต้อง และความน่าเชื่อถือในกระบวนการทำงาน และมักถูกใช้งานอย่างแพร่หลายในกระบวนการทำงานเชิงอุตสาหกรรมหรือกระบวนการทำงานที่มีการทำงานซ้ำซ้อน (สถาบันยานยนต์, 2559) สำหรับการช่วยลดต้นทุน เพิ่มประสิทธิภาพในกระบวนการทำงาน อีกทั้งยังเป็นการส่งผลดีแก่ผู้รับบริการปลายทางที่ได้ใช้งานวัตถุดิบหรือผลิตภัณฑ์ได้เร็วมากขึ้นจากการทำงานของระบบอัตโนมัติ สามารถใช้เป็นจุดเด่น

และสร้างข้อได้เปรียบทางการแข่งขันจากการนำเอาเทคโนโลยีเข้ามาใช้หรือมีส่วนร่วมในการปฏิบัติงาน (แผนกนโยบายและแผนฝ่ายยุทธศาสตร์อุตสาหกรรม สถาบันไฟฟ้าและอิเล็กทรอนิกส์, 2560) ซึ่งสามารถแบ่งระบบอัตโนมัติออกเป็น 4 รูปแบบ ได้แก่ 1) ระบบที่คิดเหมือนมนุษย์ (Systems that think like humans) คือ กลไกของกิจกรรมที่เกี่ยวข้องกับความคิดมนุษย์ เช่น การตัดสินใจ การแก้ปัญหา หรือการเรียนรู้ เป็นต้น 2) ระบบที่กระทำเหมือนมนุษย์ (Systems that act like humans) การกระทำเหมือนมนุษย์ เช่น สื่อสารได้ด้วยภาษาที่มนุษย์ใช้ อาทิ การแปลงข้อความเป็นคำพูด และการแปลงคำพูดเป็นข้อความ มีประสาทรับสัมผัสคล้ายมนุษย์ เช่น คอมพิวเตอร์รับภาพได้โดยอุปกรณ์รับสัมผัส แล้วนำภาพไปประมวลผล เคลื่อนไหวได้คล้ายมนุษย์ เช่น หุ่นยนต์ช่วยงานต่างๆ รวมถึงสามารถเรียนรู้ได้ โดยสามารถตรวจจับรูปแบบการเกิดของเหตุการณ์แล้วปรับตัวสู่สิ่งแวดล้อมที่เปลี่ยนไปได้ 3) ระบบที่คิดอย่างมีเหตุผล (Systems that think rationally) คิดอย่างมีเหตุผล หรือคิดถูกต้อง เช่น ใช้หลักตรรกศาสตร์ในการคิดหาคำตอบอย่างมีเหตุผล เช่น ระบบผู้เชี่ยวชาญ เป็นต้น และ 4) ระบบที่กระทำอย่างมีเหตุผล (Systems that act rationally) กระทำอย่างมีเหตุผล เช่น โปรแกรมที่มีความสามารถในการกระทำหรือเป็นตัวแทนในระบบอัตโนมัติต่างๆ สามารถกระทำอย่างมีเหตุผลเพื่อบรรลุเป้าหมายที่ได้ตั้งไว้ เช่น เอเจนต์ในระบบขับรถอัตโนมัติ ที่มีเป้าหมายว่าต้องไปถึงเป้าหมายในระยะทางที่สั้นที่สุด ต้องเลือกเส้นทางที่ไปยังเป้าหมายที่สั้นที่สุดที่เป็นไปได้ จึงจะเรียกได้ว่า เอเจนต์กระทำอย่างมีเหตุผล รวมถึงเอเจนต์ในเกมหมากรุก ที่มีเป้าหมายว่าต้องเอาชนะคู่ต่อสู้ ต้องเลือกเดินหมากที่สามารถทำให้คู่ต่อสู้แพ้ได้ เป็นต้น (พรจิต ประทุมสุวรรณ, 2561) โดยมีแนวทางการประยุกต์ใช้ระบบอัตโนมัติ ดังนี้

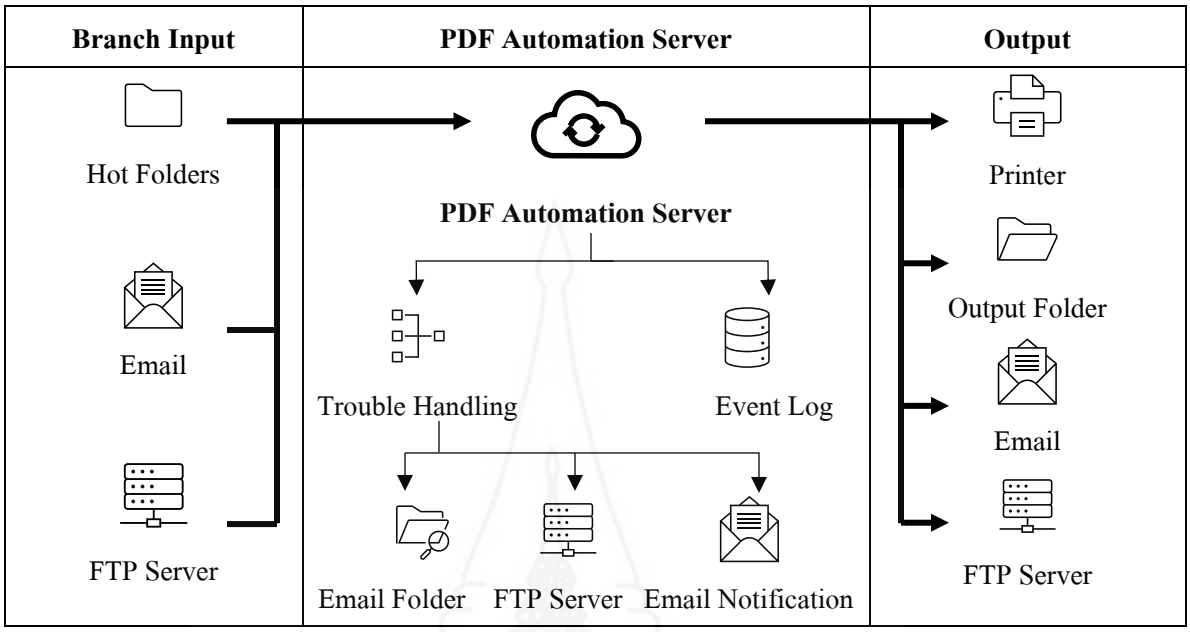
1.1.1 ระบบออกแบบอัตโนมัติ เป็นระบบอัตโนมัติในการระบบการคิดเหมือนมนุษย์ เพื่อช่วยในการออกแบบผลิตภัณฑ์อย่างใดอย่างหนึ่งให้เป็นต้นแบบ สำหรับใช้ในการนำเสนอหรือพิจารณาก่อนตัดสินใจทำการลงมือผลิตเป็นผลิตภัณฑ์จริงขึ้น โดยอาศัยโมดูลเป็นตัวช่วยในการทำหน้าที่วิเคราะห์ตามรูปแบบเทคนิคที่กำหนดขึ้นให้แสดงผลออกมาเป็น 2D หรือ 3D ในรูปแบบต่างๆ เพื่อให้ผู้ใช้งานได้ใช้ประกอบการตัดสินใจในการสั่งการผลิต (สมลักษณ์ วรรณฤมล กีเยลาโรว่า, ประภัสสร ประดุจพงษ์เพชร, และ โชคนิธิ นาคเมธี, 2561)

1.1.2 แขนกลจับชิ้นงานจากเครื่องคัดแยกวัสดุอัตโนมัติตามสายพาน เป็นระบบอัตโนมัติที่พัฒนาขึ้นเพื่อใช้ในการคัดแยกวัสดุโดยอาศัยหลักตรรกศาสตร์วัดขนาดของวัสดุที่กำหนดขึ้น และหลักการขับเคลื่อนด้วยระบบนิวแมติกส์ ซึ่งทำให้แขนกลเป็นระบบอัตโนมัติที่มีการใช้เหตุผลในการตัดสินใจและกระทำตามเงื่อนไขที่ถูกกำหนดขึ้น (ปิยะวัฒน์ ศรีธรรม, 2561)

1.2 ออโตเมชันเซิร์ฟเวอร์ (Automation Server)

ออโตเมชันเซิร์ฟเวอร์ (Automation Server) เป็นแนวคิดในการผสานระหว่างกระบวนการทำงานของระบบอัตโนมัติให้ทำงานตามความต้องการของผู้ใช้งานกับการทำงานของเครื่องแม่ข่าย เพื่อทำให้ผลลัพธ์ของกระบวนการทำงานนั้นเกิดประสิทธิภาพในการทำงาน มีความคุ้มค่า รวดเร็ว รวมถึงลดหน้าที่การทำงานของมนุษย์และเกิดความผิดพลาดน้อยที่สุด โดยมุ่งเน้นให้ผลลัพธ์หรือผลิตภัณฑ์นำออกไปสู่ผู้ใช้งานปลายทางตามที่คาดหวังไว้ ทั้งในด้านซอฟต์แวร์ และด้านการประยุกต์ใช้ (Microsoft, 2559) ซึ่งออโตเมชันเซิร์ฟเวอร์ที่ถูกนำมาใช้งานในปัจจุบันนำมาใช้งานทางด้านการสร้างหรือประมวลผลให้เกิดข้อมูลเพื่อส่งต่อไปยังผู้ใช้งานแบบอัตโนมัติ เช่น

1.2.1 PDF Automation Server ซึ่งเป็นระบบแม่ข่ายอัตโนมัติที่มีการออกแบบกระบวนการทำงานตามที่กำหนดไว้เพื่อประมวลผลเป็น PDF File สำหรับส่งต่อไปยังหน่วยการทำงานอื่นๆ ที่มีการรับค่าข้อมูลจากแหล่งข้อมูลต่างๆ เช่น โฟลเดอร์ อีเมล์ หรือเครื่องแม่ข่ายสั่งงานอื่นๆ เป็นต้น โดยเมื่อได้รับคำสั่งการทำงานจะทำการส่งคำสั่งไปยังเครื่องแม่ข่ายอัตโนมัติที่ทำหน้าที่ประมวลผลเป็น PDF File ด้วยกรรมวิธีการจัดการต่างๆ บนเครื่องแม่ข่ายและทำการจัด PDF File ที่ผ่านการประมวลผลไปทำการจัดเก็บไว้ตามแหล่งพักข้อมูลปลายทาง รวมถึงการจัดเก็บประวัติการสั่งงานและการประมวลผลไว้ เพื่อให้เครื่องลูกข่ายสามารถดึงข้อมูลไปใช้ผ่านการสั่งงานด้วยรูปแบบวิธีการต่างๆ เช่น สั่งพิมพ์ ส่งจดหมายอิเลคทรอนิกส์ จัดเก็บเป็นแฟ้มข้อมูล หรือแม้แต่ส่งต่อไปยังแม่ข่ายอื่น เป็นต้น ดังภาพที่ 2.1



ภาพที่ 2.1 PDF Automation Server

(Qoppa Software, 2563)

1.2.2 ระบบแม่ข่ายอัตโนมัติเพื่อศูนย์กลางข้อมูล เป็นแพลตฟอร์มการจัดการโครงสร้างพื้นฐานแบบครบวงจรที่ทำงานโดยอัตโนมัติและจัดเตรียมกระบวนการสำหรับการจัดเตรียมการแก้ไขและการปฏิบัติตามฐานข้อมูลและมิดเดิลแวร์ รวมถึงยังให้การวิเคราะห์ที่มีประสิทธิภาพ เพื่อช่วยลดต้นทุนการดำเนินงานด้านไอทีและเร่งการส่งมอบบริการโครงสร้างพื้นฐานที่ปลอดความเสี่ยงได้ ทั้งนี้รูปแบบการทำงานของระบบแม่ข่ายอัตโนมัติอาจประกอบไปด้วย

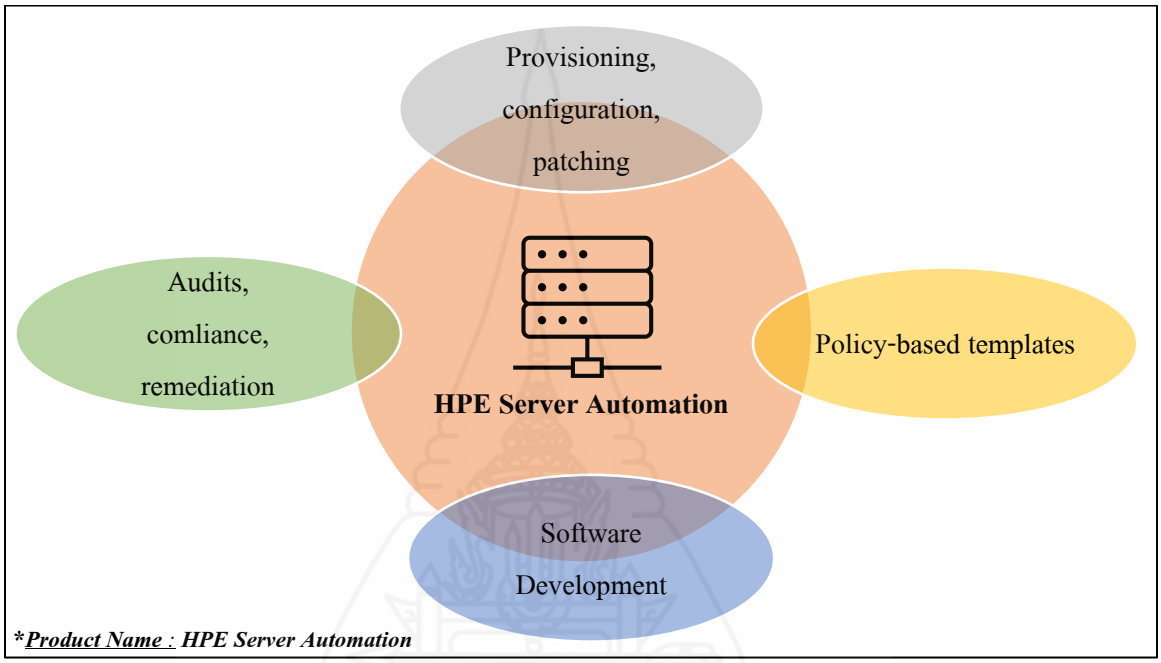
1) การกำหนดค่าคำสั่งงานในทางธุรกิจเพื่อใช้เป็นโครงสร้างพื้นฐานการทำงานในกระบวนการต่างๆ

2) การกำหนดรูปแบบนโยบายการดำเนินการที่เป็นพื้นฐานในการทำงาน เพื่อใช้เป็นมาตรฐานลำดับการทำงานในกระบวนต่อไปตามลำดับ

3) การพัฒนาซอฟต์แวร์เพื่อเติมเต็มส่วนการทำงานที่ขาดให้เติมเต็มในบริบทของการทำงาน

4) การตรวจสอบเพื่อนำไปสู่การแก้ไขก่อนนำไปใช้งานจริง

โดยรูปแบบการทำงานของระบบแม่ข่ายอัตโนมัติที่ได้กล่าวข้างต้นนั้น เป็นตัวอย่างของการทำงานในระบบแม่ข่ายอัตโนมัติในผลิตภัณฑ์ HPE Server Automation ดังภาพที่ 2.2



ภาพที่ 2.2 Data Center Automation

(LearnIT, 2563)

2. กระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง (CI/CD)

กระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง (Continuous Integration - Continuous Delivery and Deploy : CI/CD) เป็นแนวคิด ที่มาจากการนำความหมายของ Continuous Integration มารวมเข้ากับ Continuous Delivery หรือบางแหล่งที่มามีการให้ความหมายของ Continuous Delivery ควบรวมเข้ากับ Continuous Delivery and Deployment เมื่อมีการนำมาใช้กับการทำงานแบบอัตโนมัติด้านการนำผลิตภัณฑ์ซอฟต์แวร์ออกสู่การใช้งานจริง ความหมายของ Continuous Integration หรือเรียกภายใต้การย่อคำว่า CI คือ การนำกระบวนการการทำงานตามแบบแผนที่ได้กำหนดไว้ผนวก

รวมร่วมกับการประยุกต์ใช้เทคโนโลยีต่างๆให้กระบวนการการทำงานที่กำหนด สามารถทำงานได้อย่างต่อเนื่องภายใต้นิยามความหมายของ Digital 4.0 (Machine to Machine) ที่มุ่งเน้นให้เกิดการสื่อสารกันให้เกิดกระบวนการทำงานระหว่างเครื่องจักรและเครื่องจักรในรูปแบบการทำงานอัตโนมัติ อาทิ การทำงานของหุ่นยนต์สายพานในอุตสาหกรรมการผลิตที่ส่งต่อการทำงานจากการบรรจุภัณฑ์ไปยังการติดฉลากและส่งต่อไปทำงานยังส่วนต่างๆ จนถึงขั้นตอนการขนส่ง หรือกระบวนการทำงานระหว่างเครื่องคอมพิวเตอร์ที่เกิดการสั่งการจากเครื่องหนึ่งและทำงานต่อเนื่องไปยังอีกเครื่องหนึ่งได้ โดยมนุษย์จะถูกลดบทบาทในการทำงานแต่ทำหน้าที่ในการควบคุมแทนซึ่ง Digital 4.0 (Machine to Machine) มีวัตถุประสงค์เพื่อเพิ่มความสามารถในการทำงานที่รวดเร็ว สะดวก และถูกต้อง อีกทั้งยังช่วยลดบทบาทและหน้าที่ของมนุษย์บางส่วนที่อาจทำให้เกิดข้อผิดพลาดขึ้นได้ภายในกระบวนการทำงาน ความหมายของ Continuous Delivery หรือ Continuous Delivery and Deployment หรือเรียกภายใต้การย่อคำว่า CD คือ กระบวนการการทำงานแบบต่อเนื่องที่เน้นการทำงานแบบอัตโนมัติ โดยอาศัยรูปแบบการทำงานของ Continuous Integration เพื่อมุ่งหวังให้การทำงานใดๆตามกระบวนการ สามารถทำงานแบบส่งต่อระหว่าง Machine to Machine ได้ต่อเนื่อง และเมื่อผ่านกระบวนการสุดท้ายจะเกิดเป็นผลิตภัณฑ์หนึ่งที่สามารถนำไปใช้งานได้อย่างมีประสิทธิภาพ

ดังนั้น Continuous Integration and Continuous Delivery and Deployment คือกระบวนการทำงานที่อาศัยเทคโนโลยีเพื่อมุ่งเน้นให้เทคโนโลยีสามารถทำงานได้ตามกระบวนการใดกระบวนการหนึ่งที่กำหนดขึ้นได้อย่างอัตโนมัติสู่การเกิดผลิตภัณฑ์ที่สามารถนำไปใช้งานได้อย่างมีประสิทธิภาพบนพื้นฐานการทำงานตามหลัก Digital 4.0 ที่สามารถทำงานได้ระหว่างเครื่องและเครื่อง หรือ Machine to Machine เพื่อลดข้อผิดพลาดที่เกิดขึ้นของมนุษย์ในระหว่างกระบวนการทำงาน ทำให้เกิดความสะดวกและรวดเร็วในการนำไปใช้งานและช่วยเพิ่มผลผลิตจากการนำไปประยุกต์ใช้

2.1 นิยามแนวคิด

Tsalolikhin (2561) กล่าวว่า CI เป็นแนวทางการพัฒนาซอฟต์แวร์ที่สมาชิกของทีมรวมการทำงานบ่อยครั้งโดยแต่ละคนจะรวมงานอย่างน้อยทุกวัน การรวมงานแต่ละครั้งจะถูกตรวจสอบด้วยการสร้างการทำงานอัตโนมัติ เพื่อตรวจหาข้อผิดพลาดการรวมได้เร็วที่สุด ซึ่งหลายทีมพบว่าวิธีการนี้นำไปสู่การลดปัญหาการทำงานรวมกันเป็นอย่างมาก และยังช่วยให้ทีมพัฒนาซอฟต์แวร์ที่มีทำงานได้รวดเร็วยิ่งขึ้น ในส่วนของ CD เป็นการปรับการเปลี่ยนแปลงทุกอย่างให้เกิดการทำงานโดย

อัตโนมัติ สู่การส่งมอบอย่างต่อเนื่อง โดยผ่านการบูรณาการอย่างต่อเนื่องสู่การผลิตซอฟต์แวร์ที่ผ่านการทดสอบอัตโนมัติ

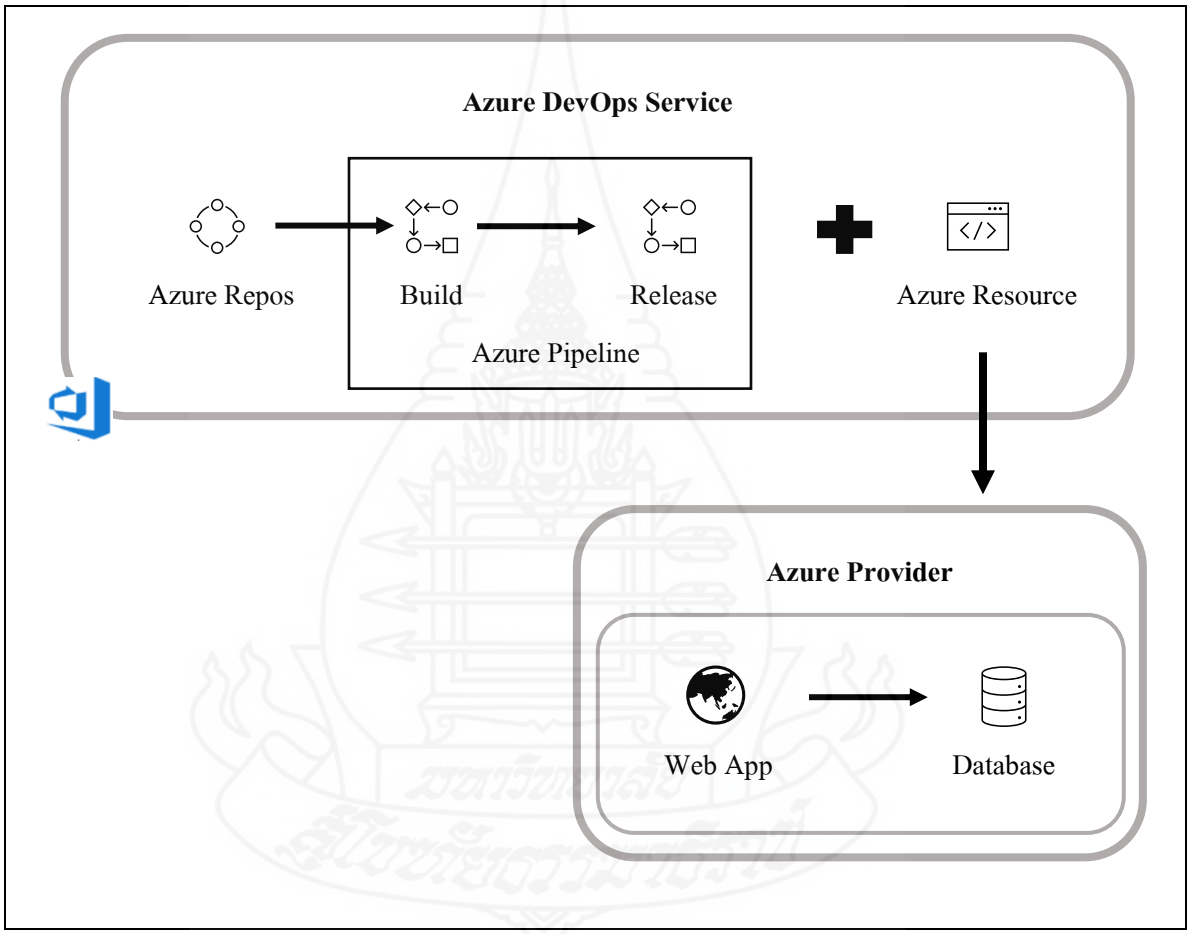
Arachchi (2561) กล่าวว่า CI เป็นการพัฒนาซอฟต์แวร์ที่สมาชิกในทีมรวมงานกันเป็นประจำโดยอัตโนมัติ โดยอาจใช้สร้างการทดสอบและตรวจสอบ เพื่อช่วยในการค้นหาและแก้ไขข้อบกพร่องได้อย่างรวดเร็วและช่วยปรับปรุงคุณภาพของซอฟต์แวร์ สำหรับ CD เป็นความสามารถในการเปลี่ยนแปลงทุกประเภทรวมถึงคุณสมบัติใหม่ให้เกิดการเปลี่ยนแปลงการกำหนดค่าการแก้ไขข้อบกพร่องและการทดลองเข้าสู่การสร้างให้นำไปสู่การใช้งานของผู้ใช้อย่างปลอดภัยและรวดเร็วในวิธีการที่ยั่งยืนโดยเปลี่ยนแปลงกระบวนการทุกอย่างที่เกิดขึ้นสู่การทำงานแบบอัตโนมัติ

DOAN and TRINH (2559) กล่าวว่า CI เป็นกระบวนการช่วยให้สามารถปรับใช้ซอฟต์แวร์ได้ทุกที่ทุกเวลา คุณลักษณะนี้มีค่าอย่างยิ่งสำหรับลูกค้าเนื่องจากนำไปสู่การนำไปใช้ของผลิตภัณฑ์จริงที่ใช้งานได้ และมีคุณภาพพร้อมที่จะใช้งานและส่งมอบให้ผู้ใช้ CD เป็นการแก้ไขปัญหาโดยการกำหนดค่าสภาพแวดล้อมการทำงานโดยอัตโนมัติให้ซอฟต์แวร์พร้อมที่จะนำไปใช้งานได้ทุกที่ทุกเวลาอย่างแท้จริง การทำงานนี้สนับสนุนอย่างมากในการทำงานรูปแบบ Agile และโดยทั่วไป

2.2 การนำไปใช้

กระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง (CI/CD) มักนิยมนำไปใช้ในการออกแบบและพัฒนาเป็นแพลตฟอร์มการทำงานแบบแม่ข่ายเสมือนจริง (Virtualization Server) ซึ่งส่วนใหญ่จะออกมาในรูปแบบของ Cloud Server ซึ่งเป็นขั้นตอนในการส่งมอบชุดคำสั่งการทำงานเข้าสู่กระบวนการบูรณาการและส่งมอบเข้าสู่การนำไปใช้งานบน Cloud Server ในรูปแบบของ Package การทำงานที่จำเป็นต้องใช้งานตาม Package ที่มีการกำหนด เช่น Azure Cloud Server, Google Cloud Server หรือ Amazon Web Services เป็นต้น ด้วยข้อจำกัดในการเข้าถึงกระบวนการทำงานของเครื่องแม่ข่าย กระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่องจึงเป็นวิธีการที่ผู้ให้บริการ Cloud Server นำมาใช้ในการอำนวยความสะดวกแก่ผู้ใช้งานในการส่งผลิตภัณฑ์ด้านซอฟต์แวร์ให้แก่ผู้ใช้งานปลายทาง โดยผ่านกระบวนการบูรณาการที่นำซอฟต์แวร์ผ่านกลไกอัตโนมัติตั้งแต่กระบวนการคอมไพล์และดำเนินการตาม Workflow ที่กำหนดขึ้นตามความต้องการสู่กระบวนการการส่งมอบและปรับใช้แบบต่อเนื่องบนสภาพแวดล้อมการทำงานจริง (สนิธพรรณ จิตตั้งสมบูรณ์, 2559) ดังเห็นได้ในกระบวนการของ Azure Cloud Server ที่มีกระบวนการ

ทำงานด้วย Azure DevOps Service ในการดึงชุดคำสั่งการทำงานจากแหล่งเก็บชุดคำสั่งในลำดับชั้นที่กำหนดค่าไว้นำไปเข้าสู่กระบวนการสร้างและจัดการการเผยแพร่ตามการทำงานที่สั่งการด้วย Azure Pipeline รวมกับ Azure Resource หรือองค์ประกอบต่างๆ ที่มีความจำเป็นต่อการทำงาน เพื่อการส่งต่อไปยัง Azure Provider ที่ทำหน้าที่ในการให้บริการผลิตภัณฑ์ซอฟต์แวร์สู่กระบวนการการติดตั้งซอฟต์แวร์และฐานข้อมูลให้สมบูรณ์พร้อมนำไปใช้งานได้จริง ตามภาพที่ 2.3



ภาพที่ 2.3 กระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่องบน Azure Server

(Ricardo Humberto Vivanco Effio, 2562)

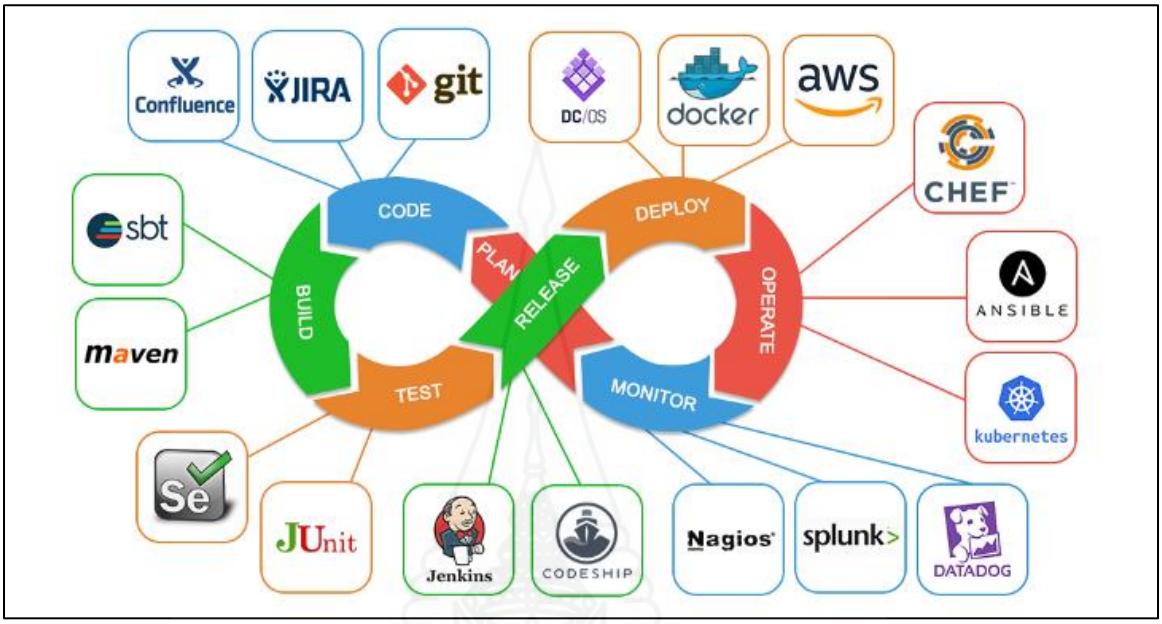
3. ขั้นตอนการพัฒนาและปฏิบัติการ (Development and Operations)

ขั้นตอนการพัฒนาและปฏิบัติการ (Development and Operations) เป็นขั้นตอนในการทำงานตามแนวคิดการทำงานระหว่างฝ่ายพัฒนาและฝ่ายปฏิบัติการ เพื่อลดข้อขัดแย้งจากปัญหาต่างๆ ในการทำงาน เช่น การติดต่อสื่อสารหรือความผิดพลาดระหว่างการทำงาน เป็นต้น อันทำให้เกิดผลเสียในการปฏิบัติงาน ตั้งแต่การส่งมอบล่าช้า และทำให้เกิดปัญหาต่อผู้ใช้งานที่ไม่สามารถใช้งานฟีเจอร์การทำงานใหม่ได้ตามกำหนดเวลา แนวคิดขั้นตอนการพัฒนาและปฏิบัติการจึงเกิดขึ้นโดยมีเป้าหมายในการส่งมอบการปรับปรุงซอฟต์แวร์ใหม่อย่างต่อเนื่องโดยอัตโนมัติ ซึ่งจะต้องมีความถูกต้องและความน่าเชื่อถือ (Leonardo Alexandre Ferreira Leite, Carla Aguiar, Fabio Kon, Dejan Milojicic, & Paulo Meirelles, 2562)

โดยในขั้นตอนการพัฒนาและปฏิบัติการ มักถูกเรียกในเชิงเทคนิคด้วยชื่อแบบสั้นในรูปแบบภาษาอังกฤษว่า เดฟออฟ (DevOps) โดย Dev มาจากคำว่า Development ที่มีความหมายถึงกระบวนการพัฒนาหรือขั้นตอนการพัฒนา อันมีกระบวนการที่แตกต่างกันไป ขึ้นอยู่กับบริบทของงานแต่ละหน่วยที่มีการปรับปรุงเปลี่ยนแปลงกระบวนการพัฒนานั้นให้มีความเหมาะสมกับขั้นตอนในการทำงานอันนำไปสู่การปฏิบัติการ หรือ Ops ที่จะกล่าวโดย Ops มาจากคำว่า Operations ที่ให้ความหมายถึงขั้นตอนในการปฏิบัติการ ขั้นตอนการปฏิบัติงาน รวมไปถึงลำดับงานหรือ เหตุการณ์อันนำไปสู่การบรรลุเป้าหมายที่ได้ถูกกำหนดไว้ โดยเมื่อถูกนำมาผสานรวมกันเป็นขั้นตอนที่เชื่อมโยงกันจะสามารถช่วยลดระยะเวลาในการปฏิบัติงานต่างๆ ให้เป็นไปตามจุดประสงค์หลักของ DevOps กล่าวคือ การผลักดันให้เกิดกระบวนการที่สามารถทำงานร่วมกันระหว่าง Development และ Operations ให้มีประสิทธิภาพ รวดเร็ว และมีความน่าเชื่อถือในการทำงาน แต่ไม่ได้หมายรวมถึงกระบวนการอัตโนมัติ หากแต่กระบวนการอัตโนมัตินั้นเกิดขึ้นจากขั้นตอนการพัฒนาที่ถูกปรับปรุงภายใต้กระบวนขั้นตอนพัฒนาและปฏิบัติการ เพื่อให้เกิดผลลัพธ์ที่รวดเร็วตามวัตถุประสงค์ใดวัตถุประสงค์หนึ่ง (Sangakong A., 2563)

ในการประยุกต์ขั้นตอนการพัฒนาและปฏิบัติการ มักถูกนำมาประยุกต์ตามบริบทการทำงานและปรับเปลี่ยนให้เกิดความสอดคล้องและเหมาะสมกับการนำไปใช้งาน เพื่อแก้ไขปัญหาที่เกิดขึ้น ซึ่งการพัฒนาส่วนหนึ่งสามารถนำเอาซอฟต์แวร์ต่างๆ ที่เกิดขึ้นหรือมีอยู่แล้วร่วมกับเครื่องมือหรืออุปกรณ์ที่มีอยู่มาทำการปรับใช้ให้เกิดประโยชน์ต่อหน่วยงานให้สูงที่สุด โดยส่วนมากนิยมนำมาใช้ในการพัฒนาอยู่บนพื้นฐานของแหล่งข้อมูลแบบเปิด (Open Source) อันสามารถนำมาปรับใช้

และประยุกต์ให้เกิดลักษณะการทำงานที่หลากหลายเป็นไปตามวัฒนธรรมขององค์กรนั้นๆ ดังภาพที่ 2.4



ภาพที่ 2.4 แนวทางการประยุกต์ขั้นตอนการพัฒนาและปฏิบัติการ : DevOps (Singh J., 2561)

ขั้นตอนการพัฒนาและปฏิบัติการ จึงถูกเลือกนำมาปรับปรุงและปรับใช้ให้เกิดเป็นซอฟต์แวร์ในการช่วยอำนวยความสะดวกให้เหมาะสมกับองค์กรและหน่วยงานในรูปแบบต่างๆ ซึ่งหนึ่งในรูปแบบในการพัฒนาให้สอดคล้องกับการแก้ปัญหาในองค์กร อันเป็นไปตามระบบอัตโนมัติที่ถูกกล่าวไว้ตามวรรณกรรมข้างต้นที่เกี่ยวข้องอันได้แก่ ระบบอัตโนมัติและออโตเมชันเซิร์ฟเวอร์ กระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง (CI/CD) และขั้นตอนการพัฒนาและปฏิบัติการ ตามขั้นตอนดังนี้

3.1 วางแผน (Plan) เนื่องจากหน่วยงานมีการกำหนดข้อตกลงเพื่อสนับสนุนระดับการบริการทางธุรกิจ ในการกำหนดระยะเวลาในการทำงานหรือแก้ไขปัญหาเมื่อพบปัญหาหรือได้รับแจ้งให้แล้วเสร็จตามกำหนดเวลา ซอฟต์แวร์ที่พบปัญหาจึงต้องถูกแก้ไขปัญหาให้เป็นไปตามข้อตกลงดังกล่าว เพื่อสนับสนุนการบริการทางธุรกิจให้แก่ผู้ใช้งานและลูกค้า แต่มักพบปัญหาเรื่องความล่าช้าในการนำซอฟต์แวร์ที่ถูกแก้ไขนำไปใช้งานบนเครื่องแม่ข่าย

3.2 การเขียนโปรแกรม (Code) เป็นการพัฒนาและแก้ไขปัญหาหลังขั้นตอนการวางแผน เพื่อการนำชุดคำสั่งไปใช้ในการพัฒนาซอฟต์แวร์ หรือนำชุดคำสั่งที่มีการพัฒนาขึ้นเข้าสู่กระบวนการการแก้ไขตามแผนที่วางไว้ ก่อนนำเข้าสู่กระบวนการสร้างชุดทำงานโปรแกรม (Build) และนำไปใช้งาน โดยจำเป็นต้องมีการทดสอบแบบหน่วยทดสอบ (Unit Test) ก่อนส่งเข้าสู่กระบวนการต่อไป เพื่อตรวจสอบความถูกต้องของการพัฒนาเบื้องต้นว่าสามารถทำงานได้ตามปกติและไม่ส่งผลกระทบต่อการทำงานอื่นในระบบหรือซอฟต์แวร์ที่พัฒนาหรือแก้ไข ซึ่งจำเป็นต้องตรวจสอบความเรียบร้อยก่อนทำการจัดเก็บชุดคำสั่งที่ได้พัฒนาและแก้ไขโดยสมบูรณ์เข้าสู่กระบวนการจัดเก็บชุดคำสั่งบน Source Control อันถือเป็นการเริ่มต้นการทำงานตามแนวคิดกระบวนการบูรณาการ Continuous Integration : CI (iFew in Technology, 2561) ตั้งแต่ขั้นตอนทำการ Push Code เข้าสู่ Source Control

3.3 การสร้างชุดทำงานโปรแกรม (Build) แนวทางการสร้างชุดทำงานโปรแกรมที่สามารถนำไปใช้กับแนวคิดกระบวนการบูรณาการนิยมใช้การทำงานจากการสร้างตัว Trigger การทำงานไปยังเครื่องมือจัดการการบูรณาการ เพื่อสร้างพื้นที่การปฏิบัติงานของชุดคำสั่งจากขั้นตอนการเขียนโปรแกรม และทำงานตามการกำหนดเส้นทางการทำงาน (Workflow) ที่พัฒนาขึ้นด้วยรูปแบบของคำสั่งภาษาเครื่องและภาษาสื่อสารระหว่างเครื่อง โดยอาศัยส่วนเชื่อมประสานของตัวสร้างชุดทำงานเพื่อทำการคอมไพล์ (Complier) และบีบอัดเพื่อสร้างชุดคำสั่งโปรแกรมสำเร็จ (Package)

3.4 การทดสอบ (Test) เป็นกระบวนการที่ถูกปรับเปลี่ยนเพื่อการทำงานที่เหมาะสมตามบริบทของหน่วยงาน ซึ่งแต่ละหน่วยงานได้มีบริบทการทดสอบที่แตกต่างกัน โดยปรับให้สามารถยืดหยุ่นได้ด้วยการทดสอบแบบแมนวล โดยอาศัยผู้ทำหน้าที่ทดสอบระบบมาใช้ในการทดสอบ หรืออาศัยระบบอัตโนมัติที่จำเป็นต้องพัฒนาขึ้นด้วยผู้พัฒนาระบบทดสอบอัตโนมัติเพื่อปรับเปลี่ยนเป็นกระบวนการทดสอบแบบอัตโนมัติ โดยสามารถเกิดการทดสอบได้ทั้งก่อน (Pre-Test) หรือหลัง (Post Test) จากที่ได้นำขึ้นซอฟต์แวร์เพื่อใช้งาน หรือสามารถเกิดการทดสอบขึ้นได้ทั้ง 2 เหตุการณ์

3.5 กระบวนการเผยแพร่ (Release) ถือเป็นกระบวนการของการเริ่มต้นแนวคิดการส่งมอบและปรับใช้แบบต่อเนื่อง Continuous Delivery and Deploy : CD (Aleksey Tsalolikhin, 2561) โดยกระบวนการนี้เป็นการจัดเตรียมชุดคำสั่งที่เกิดขึ้นจากขั้นตอนการสร้างชุดทำงานโปรแกรมเพื่อเข้าสู่การจัดเวอร์ชันของซอฟต์แวร์ที่พร้อมนำออกเผยแพร่ และปล่อยออกขึ้นสู่

ขั้นตอนการติดตั้ง (Deploy) เพื่อนำไปใช้งานจริงบนเครื่องแม่ข่ายหรือแหล่งให้บริการอื่น เช่น App Store, Play Store หรือศูนย์รวมซอฟต์แวร์เพื่อการนำไปใช้งาน เป็นต้น

3.6 การติดตั้งและนำไปใช้งานจริง (Deploy) เพื่อให้เป็นไปตามข้อตกลงระดับการบริการที่ถูกกำหนด ในขั้นตอนนี้จะต้องเสร็จสิ้นทันตามระยะเวลาที่ถูกกำหนดไว้ ซึ่งในขั้นตอนนี้เป็นขั้นตอนที่นำซอฟต์แวร์หรือข้อปัญหาที่ได้รับการแก้ไขนำออกสู่ผู้ใช้งานหรือลูกค้าให้สามารถใช้งานได้ตามปกติและต้องอยู่ภายในระยะเวลาที่เป็นไปตามข้อตกลงระดับบริการจึงจะสำเร็จผลตามที่หน่วยงานคาดหวัง และถือเป็นความสำเร็จในระดับหน่วยงานและระดับองค์กรในภาพรวม และสร้างความพึงพอใจให้แก่ผู้ใช้งานเมื่อได้รับการใช้งานตามการพัฒนาและการแก้ไข

3.7 การปฏิบัติการ (Operate) เพื่อการตรวจสอบและติดตามปัญหาซึ่งเกิดขึ้นด้วยการตรวจสอบแบบบันทึกผลการทำงานภายในเครื่องมือการตรวจสอบ และแบบบันทึกผลการติดตามการทำงานของเครื่องแม่ข่ายซึ่งเป็นส่วนของการทำงานโดยทีมปฏิบัติการที่ทำหน้าที่ในการตรวจสอบและแก้ไขปัญหา เพื่อให้เครื่องแม่ข่ายสามารถทำงานได้อย่างถูกต้อง

4. ข้อตกลงระดับการบริการ (Service Level Agreements)

4.1 นิยามแนวคิด

สมศักดิ์ ขาวสุวรรณ (2561) กล่าวว่า ข้อตกลงระดับการบริการทางธุรกิจ หมายถึง เป็นเอกสารบันทึกข้อตกลงระหว่างผู้ให้บริการและผู้ใช้บริการ ซึ่งเปรียบเสมือนคำสัญญาถึงระดับคุณภาพของบริการที่ทั้งสองฝ่ายยอมรับได้ โดยข้อตกลงควรประกอบด้วยขอบเขตการให้บริการ ข้อกำหนดการให้บริการระดับการให้บริการ ขั้นตอนการให้บริการและการรับฟังความคิดเห็นของการรับบริการ

องค์การพิพิธภัณฑ์วิทยาศาสตร์แห่งชาติ (2561) กล่าวว่า ข้อตกลงระดับการบริการทางธุรกิจ หมายถึง ข้อตกลงหรือสัญญา การให้บริการระหว่างผู้ให้บริการและผู้รับบริการเพื่อให้บริการในระดับที่ทั้งสองฝ่ายตกลงกัน โดยจัดทำขึ้นเป็นลายลักษณ์อักษรและมีผลบังคับใช้ ภายในเวลาและเงื่อนไขที่กำหนด

กองวิเคราะห์และตรวจสอบทรัพยากรธรณี กรมทรัพยากรธรณี (2562) กล่าวว่า ข้อตกลงระดับการให้บริการ หมายถึง เอกสารที่บันทึกข้อตกลงระหว่างผู้ให้บริการและผู้รับบริการเป็นลายลักษณ์อักษรอย่างชัดเจน ถึงระดับการให้บริการ (Level of Service) ขั้นต่ำที่ทั้งสองฝ่ายยอมรับ

ได้ ซึ่งข้อตกลงการให้บริการเปรียบเสมือนพันธะสัญญาถึงระดับคุณภาพของบริการและความโปร่งใสที่ผู้ให้บริการให้แก่ผู้รับบริการ

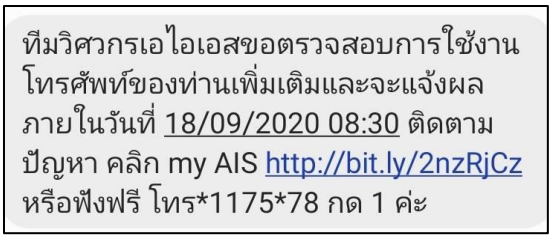
สรุปนิยามแนวคิดเกี่ยวข้องข้อตกลงระดับการบริการทางธุรกิจ หมายถึง การจัดทำข้อตกลงซึ่งจัดทำขึ้นในรูปแบบของเอกสาร ลายลักษณ์ หรือรูปแบบแผนภาพกระบวนการเพื่อเป็นข้อกำหนดในการให้บริการทางธุรกิจ อันเสมือนเป็นข้อสัญญาที่สัญญาต่อกันระหว่างฝ่ายหนึ่งกับอีกฝ่ายหนึ่ง ที่จะปฏิบัติตามขั้นตอนที่ถูกกำหนดเอาไว้สำหรับแก้ไข ปรับปรุงหรือให้บริการตามที่ได้ตกลงกัน ภายใต้ระดับการให้บริการที่กำหนดขึ้นร่วมกันทั้งสองฝ่าย ระหว่างผู้ให้บริการและผู้รับบริการ ทั้งนี้ข้อตกลงการให้บริการมักถูกนำมาใช้กับหน่วยงานทางธุรกิจเพื่อตกลงและร่วมมือกันตามขั้นตอนภายในหน่วยงาน หรือภายนอกหน่วยงาน รวมถึงลูกค้าผู้ใช้บริการ เพื่อทำให้เกิดประสิทธิภาพในการให้บริการในแต่ละส่วนให้เกิดประสิทธิภาพมากที่สุด อันจะส่งผลเสียแต่องค์กรน้อยที่สุดหากมีการปฏิบัติการตามข้อตกลงที่ได้ตกลงกันไว้

ข้อตกลงระดับการบริการ หรือ Service Level Agreements ซึ่งมักถูกเรียกใช้ในทางธุรกิจด้วยชื่อย่อว่า SLA ถือเป็นข้อกำหนด หรือข้อตกลงในระดับขั้นตอนการให้บริการ ซึ่งถูกแบ่งไว้เป็นลำดับขั้นตามสถานการณ์ที่แตกต่างกันออกไป โดยอาศัยทรัพยากรที่มีในการให้บริการอย่างจำกัดเพื่อรักษาระดับต้นทุน หรือระดับบริการไว้ให้เหมาะสมอยู่ในเกณฑ์คุณภาพที่ยอมรับได้ทั้งสองฝ่ายตามระดับการให้บริการ

4.2 ระดับการให้บริการ

การกำหนดระดับการให้บริการสามารถกำหนดขึ้นได้ภายในองค์กรและสามารถยอมรับร่วมกันได้แต่ละฝ่ายที่อยู่ในขอบเขตการให้บริการ จึงต้องการอาศัยการปรึกษา วางแผนและกำหนดระดับบริการร่วมกัน ซึ่งต้องคำนึงถึงลักษณะของงานที่ให้บริการ ระยะเวลา รวมไปถึงช่องทางในการให้บริการ เพื่อสร้างหลักประกันคุณภาพในการให้บริการอย่างมีประสิทธิภาพ โดยระดับการให้บริการโดยทั่วไปมักแบ่งออกได้เป็น 3 ประเภทระดับการให้บริการ (Motadata ITSM, 2562) ดังนี้

4.2.1 ระดับการให้บริการในระดับลูกค้า ซึ่งเป็นข้อตกลงที่กำหนดถึงการรักษาระดับคุณภาพในการให้บริการ ที่เน้นกลุ่มผู้รับบริการเป็นลูกค้าที่ส่งผลกระทบต่อธุรกิจ อาทิ ข้อกำหนดในการให้บริการลูกค้าเมื่อลูกค้าประสบการณ์ปัญหาในการใช้บริการ และสามารถแก้ไขปัญหาที่เกิดขึ้นแก่ลูกค้าได้สำเร็จภายในระดับเวลาที่กำหนด



ภาพที่ 2.5 ตัวอย่างข้อกำหนดระดับการบริการทางธุรกิจในระดับลูกค้า (AIS, 2563)

4.2.2 ระดับการให้บริการในระดับองค์กร เป็นการกำหนดข้อกำหนดการให้บริการที่ใช้ในควบคุมคุณภาพของงานที่จะถูกส่งออกไปจากฝ่ายหนึ่งไปอีกฝ่ายหนึ่ง หรือกล่าวคือถูกนำออกไปใช้งานจริง ทั้งภายในและภายนอกองค์กร อาทิ การกำหนดระยะเวลาในการแก้ไขปัญหาของซอฟต์แวร์เมื่อพบปัญหา และแก้ไขให้เสร็จสิ้นตามเวลาที่กำหนดเพื่อให้ส่วนงานอื่นสามารถดำเนินการตามขั้นตอนที่สัมพันธ์กันไปได้อย่างไม่ติดขัด

ตารางที่ 2.1 ตัวอย่างข้อกำหนดระดับการบริการทางธุรกิจในระดับองค์กร

Service Level Agreement		
Service Level	**Description**	**Target Response**
1. Outage	SaaS server down	Immediate
2. Critical	High risk of server downtime	Within 10 minutes
3. Urgent	End-user impact initiated	Within 20 minutes
4. Important	Potential for performance impact if not addressed	Within 30 minutes
5. Monitor	Issue addressed but potentially impactful in the future	Within one business day
6. Informational	Inquiry for information	Within 48 hours

KPIs and Metrics : Availability => MTTR, Reliability => MTTF

Remark : - Ensure availability of 99.9999% during holiday season.

- Response to Service Level 6 or below can be delayed up to 24 hours during the holiday season.

(Raza, M., 2020)

4.2.3 ระดับการให้บริการในระดับบริการ เป็นการกำหนดข้อกำหนดการให้บริการที่ใช้ในควบคุมคุณภาพของงานที่กำหนดขึ้น เพื่อกลุ่มผู้รับบริการกลุ่มหนึ่งโดยเฉพาะ ซึ่งมักเป็นไปในลักษณะของการทำสัญญาให้บริการซึ่งกันและกัน อาทิ ข้อกำหนดในการตกลงการให้บริการเพื่อพัฒนา แก้ไข และบำรุงรักษาตามลำดับความสำคัญในการให้บริการของกลุ่มผู้ใช้งาน NOSTRA MAP APIs เพื่อนำความสามารถสำคัญของ NOSTRA MAP APIs ไปพัฒนาต่อยอดในธุรกิจ ดังตัวอย่างตารางที่ 2.2

ตารางที่ 2.2 ตัวอย่างข้อกำหนดระดับการบริการทางธุรกิจในระดับบริการ

NOSTRA Map Online (SLA)			
ระดับความสำคัญ	**ลักษณะของปัญหาในแต่ละความสำคัญ**	**เวลาในการ Response เบื้องต้น (Initial Response)**	**เวลาในการแก้ไขปัญหาแบบถาวร (Correction)**
1	ผู้ใช้งานไม่สามารถเรียกใช้บริการออนไลน์ที่บริษัทฯ เป็นผู้ให้บริการได้ อันเนื่องมาจาก Bug ของ โปรแกรมประยุกต์ที่บริษัทฯ เป็นผู้พัฒนา	12 ชั่วโมง	1 วันทำการ
2	- ผู้ใช้งานไม่สามารถเรียกใช้บริการออนไลน์ที่บริษัทฯ เป็นผู้ให้บริการได้ อันเนื่องมาจากความเสียหายของอุปกรณ์ ฮาร์ดแวร์ เซิร์ฟเวอร์ และเน็ตเวิร์ค - ผู้ใช้บริการขอคำปรึกษาการใช้งานซอฟต์แวร์และโปรแกรมประยุกต์	12 ชั่วโมง	2 วันทำการ
การชดเชยความเสียหาย			
ค่าชดเชย $= \frac{\text{ระยะเวลา Download ส่วนเกินรับประกัน}}{\text{ระยะเวลาที่ใช้บริการทั้งหมด}} \times$ ค่าบริการแผนที่ออนไลน์			

(NOSTRA, 2563)

4.3 ระดับการให้บริการกับมาตรฐาน ISO-27001 : 2013 หมวดความสัมพันธ์ระบบบริหารจัดการความมั่นคงปลอดภัยสารสนเทศ (Information Security Management System : ISMS)

การกำหนดระดับการให้บริการถือเป็นส่วนหนึ่งของการสร้างมาตรฐานตามมาตรฐาน ISO-27001 ซึ่งในการวิจัยครั้งนี้มาตรฐานดังกล่าวได้ถูกกำหนดขึ้นจากการกำหนดระดับให้บริการของหน่วยงานที่เลือกใช้เป็นมาตรฐาน ISO-27001 : 2013 ตามหมวดระบบบริหารจัดการความมั่นคงปลอดภัยสารสนเทศ (ISMS) ที่กำหนดให้องค์กรมีการกำหนดนโยบายความมั่นคงปลอดภัยสารสนเทศ ตัวชี้วัด รวมถึงวิธีการปรับปรุงแก้ไขเมื่อเกิดปัญหาอย่างเป็นขั้นตอนและสามารถสื่อสารให้ผู้ที่เกี่ยวข้องรับทราบถึงวิธีการแก้ไขปัญหาเมื่อพบปัญหา และทำการปรับปรุงแก้ไขปัญหาตามวิธีการขั้นตอนที่กำหนดไว้ โดยต้องมีการกำหนดสิ่งที่จำเป็นต้องแก้ไขปัญหา วิธีการแก้ไขปัญหา ทรัพยากรที่ใช้ ผู้รับผิดชอบ ระยะเวลาในการแก้ไข รวมถึงวิธีการในการประเมินผลตามตัวชี้วัดที่เป็นลายลักษณ์อักษร (บริษัท ที-เน็ต จำกัด, 2563)

5. การประเมินขนาดซอฟต์แวร์ด้วยเทคนิคมาตรฐานการประมาณโมเดล COCOMO

5.1 ความสำคัญของการประเมินขนาดซอฟต์แวร์

โมเดล COCOMO เป็นโมเดลที่มีไว้สำหรับการนำเสนอข้อมูลพื้นฐานสำคัญสำหรับการวิเคราะห์และประเมินส่วนประกอบ ขนาดซอฟต์แวร์ รวมถึงนำไปสู่การประเมินราคาของซอฟต์แวร์ เพื่อใช้เป็นข้อมูลในการตัดสินใจการจัดสรรทรัพยากรและบุคลากรที่เกี่ยวข้องในการพัฒนาซอฟต์แวร์ได้อย่างเหมาะสมกับต้นทุนค่าใช้จ่ายและกำลังคน (Kumar & Bhatia, 2014)

โดยมีการแบ่งชนิดของโมเดลเพื่อนำไปใช้ในการประเมินขนาดของซอฟต์แวร์ในรูปแบบต่างๆ ได้ 3 รูปแบบ ประกอบด้วย 1) โมเดล COCOMO ขั้นพื้นฐาน มีวัตถุประสงค์เพื่อทำการคำนวณขนาดของซอฟต์แวร์โดยคร่าวอย่างรวดเร็ว สำหรับการนำไปใช้ในการบริหารจัดการโครงการเบื้องต้น 2) โมเดล COCOMO ขั้นกลาง มีวัตถุประสงค์เพื่อประเมินไปถึงต้นทุนที่ใช้ในการขับเคลื่อนโครงการจากการพัฒนาซอฟต์แวร์และนำไปสู่ข้อมูลเชิงลึก และ 3) โมเดล COCOMO เชิงลึก เป็นรูปแบบโมเดลที่ขยายความและรายละเอียดเชิงลึกลงไปจากโมเดล COCOMO ขั้นกลาง ที่มีการวัดประสิทธิภาพและการวิเคราะห์ถึงรายละเอียดของโครงการซอฟต์แวร์ เพื่อให้เกิดผลลัพธ์คู่ขนานระหว่างการคำนวณต้นทุนที่ใช้ขับเคลื่อนโครงการและประสิทธิภาพในการดำเนินงาน

เพื่อให้เกิดผลิตภัณฑ์ และโดยทั่วไปมักนิยมใช้โมเดล COCOMO ขั้นกลางและโมเดล COCOMO เชิงลึกควบคู่กันไป (Asija S., 2020)

โมเดล COCOMO ถือเป็นตัวแปรสำคัญที่ใช้ในการแสดงถึงปริมาณของผลิตภัณฑ์ซอฟต์แวร์ ซึ่งจากการประเมินขนาดซอฟต์แวร์ด้วยโมเดล COCOMO ทำให้ได้ผลลัพธ์หลัก คือ ปริมาณทรัพยากรที่ใช้ ซึ่งกล่าวคือจำนวนทรัพยากรมนุษย์ที่ใช้ในการดำเนินการโครงการซอฟต์แวร์ และ กำหนดการหรือระยะเวลาที่ใช้ในการดำเนินโครงการ (Sommerville L., 2011) ซึ่งสามารถนำไปใช้ในการวางแผนการดำเนินโครงการให้บรรลุผลตามเป้าหมายที่วางไว้ โดยแยกลักษณะงานออกเป็นส่วนๆ เพื่อให้การดำเนินการเป็นไปตามกำหนดทีละส่วนแบ่งเป็นขั้นตอนการวางแผนโครงสร้างการพัฒนาโปรแกรมและสถาปัตยกรรมระบบ การแบ่งหน่วยการพัฒนาตลอดจนการรวมระบบการพัฒนา และการนำไปใช้งานร่วมกับโครงสร้างการปฏิบัติงานจริง (Agarwal A., 2019)

5.2 เทคนิคมาตรฐานการประมาณขนาดซอฟต์แวร์ของโมเดล COCOMO

โมเดล COCOMO สามารถอ้างอิงขนาดของซอฟต์แวร์ตามการกำหนดขนาดของ Boehm ที่สามารถแบ่งขนาดของซอฟต์แวร์เป็นแบบ 1) Organic ซึ่งเป็นโครงการซอฟต์แวร์ขนาดเล็ก ที่สมาชิกในทีมทุกคนสามารถแก้ไขปัญหาและเข้าใจปัญหาที่เกิดขึ้นได้ทั้งในอดีตและปัจจุบัน แม้ว่าสมาชิกในโครงการจะมีประสบการณ์น้อย 2) Semi-detached หรือโครงการซอฟต์แวร์ขนาดกลาง ซึ่งต้องอาศัยความรู้และประสบการณ์จากการปฏิบัติงานอยู่บ้าง แต่ไม่ถึงขั้นความเชี่ยวชาญ ลักษณะของซอฟต์แวร์มีความซับซ้อนเล็กน้อย และ 3) Embedded เป็นโครงการซอฟต์แวร์ขนาดใหญ่ที่มีความซับซ้อนสูง ซึ่งจำเป็นต้องอาศัยนักพัฒนาที่มีประสบการณ์และความสามารถเชี่ยวชาญในการพัฒนาและสร้างแบบจำลองของระบบทั้งหมด ซึ่งถือเป็นซอฟต์แวร์ขนาดระดับสูงสุด (Sommerville L., 2011)

โดยขนาดของซอฟต์แวร์ต่างๆจะสามารถประเมินได้โดยอาศัยเทคนิคการคำนวณตามซอฟต์แวร์ 3 รูปแบบดังต่อไปนี้

5.2.1 โมเดล COCOMO ขั้นพื้นฐาน (Basic Model) โดยมักวัดด้วยจำนวนของบรรทัดของการเขียนโปรแกรม (Kilo-Lines of Code) เพื่อนำมาวัดความสามารถในการทำงานของคน นำไปสู่การวัดระยะเวลาในการพัฒนาเป็นจำนวนเดือน

5.2.2 โมเดล COCOMO ขั้นกลาง (Intermediate Model) จากโมเดลขั้นพื้นฐานที่เป็นเพียงการวัดจากปริมาณบรรทัดที่ใช้ในการเขียนโปรแกรม เพื่อประเมินค่าตาม

ระบบซอฟต์แวร์ที่ไม่เท่ากัน ซึ่งในบางครั้งจำนวนบรรทัดไม่สามารถนำมาใช้ในการประเมินค่าขนาดของซอฟต์แวร์ได้ จึงจำเป็นต้องอาศัยการจำแนกประเภทต่างๆร่วมด้วย ได้แก่

1) ลักษณะของซอฟต์แวร์ โดยพิจารณาจากขอบเขตของงาน ขนาดของฐานข้อมูล และความซับซ้อนในการพัฒนา

2) ลักษณะของฮาร์ดแวร์ โดยพิจารณาจากข้อจำกัดของระบบงาน ขนาดของหน่วยความจำ เสถียรภาพของเครื่องแม่ข่าย สภาพแวดล้อมของโครงสร้างและสถาปัตยกรรม รวมถึงอายุการทำงานของฮาร์ดแวร์ที่ใช้และต้องการ

3) บุคลากร โดยพิจารณาจากความสามารถและความเชี่ยวชาญของบุคลากรภายในโครงการ อาทิ ความสามารถในการวิเคราะห์ วิศวกรรม และประสบการณ์ทั้งด้านฮาร์ดแวร์และการพัฒนาซอฟต์แวร์ เป็นต้น

4) ลักษณะของโครงการ โดยพิจารณาจากเครื่องมือที่ใช้ การประยุกต์ใช้ และระยะเวลาที่จำกัด

ซึ่งการประเมินด้วยโมเดลขั้นกลางนี้ผู้จัดการโครงการจะเป็นผู้ประเมินและวิเคราะห์การประเมินจากองค์ประกอบต่างๆที่เกี่ยวข้องทั้ง 4 ด้านเพื่อนำมาใช้ในการคำนวณหาค่า Effort Adjustment Factor (EAF) และประเมินขนาดของซอฟต์แวร์ต่อไป

5.2.3 โมเดล COCOMO เชิงลึก (Detailed Model) เป็นการประเมินถึงผลกระทบของต้นทุนในการทำงานในแต่ละขั้นตอน จากการประเมินในโมเดลขั้นกลางเพื่อแตกเป็นรายละเอียดเชิงวิศวกรรมซอฟต์แวร์แยกเป็นรายการอย่างละเอียดโดยอาจเป็นซอฟต์แวร์โมดูลตามลักษณะการทำงานของซอฟต์แวร์ และออกแบบเป็นขั้นตอนของ COCOMO โดยละเอียดดังนี้ 1) การวางแผนและข้อกำหนดงาน 2) การออกแบบระบบ 3) รายละเอียดและคำอธิบายการออกแบบระบบ 4) โมดูลและการทดสอบ 5) การรวมระบบและการทดสอบรวม 6) ต้นทุน และ 7) ฟังก์ชันและองค์ประกอบสำคัญของซอฟต์แวร์

ซึ่งโมเดลแต่ละส่วนนั้นสามารถอธิบายเป็นลักษณะตามวงจรในการพัฒนาระบบและการบริหารจัดการโครงการ หรือเป็นไปตามบริบทรูปแบบการทำงานของหน่วยงานและองค์กรที่มีการกำหนดไว้

6. งานวิจัยที่เกี่ยวข้อง

จากการศึกษาค้นคว้าวรรณกรรมที่มุ่งเน้นศึกษาในส่วนของงานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์ ได้พบงานวิจัยที่เกี่ยวข้องดังต่อไปนี้

Doan and Trinh (2559, น. 36-39, 65-68) นำเสนอการปรับใช้กระบวนการการบริหารแบบ Scrum ซึ่งเป็นส่วนหนึ่งของ Agile Methodology ที่มีประสิทธิภาพ ร่วมกับการทำงานแบบบูรณาการ และการปรับใช้การส่งมอบแบบต่อเนื่อง โดยเปรียบเทียบจากระยะเวลาและคุณภาพของงานจากการสภาพแวดล้อมที่สร้างขึ้น 2 แบบ ได้แก่ การส่งมอบแบบผ่านสื่อสังคมออนไลน์ และการส่งมอบแบบประยุกต์ใช้การทำงานแบบบูรณาการ และการปรับใช้การส่งมอบแบบต่อเนื่อง ซึ่งผลการวิจัยระบุว่า การการทำงานแบบบูรณาการ และการปรับใช้การส่งมอบแบบต่อเนื่องมีผลการทำงานที่ดีโดยมีประสิทธิภาพดีในด้านของคุณภาพและเวลา เนื่องจากสามารถใช้ลดเวลาในการส่งมอบและมีเวลาในการปรับปรุงคุณภาพของแอปพลิเคชันที่ส่งมอบได้มากกว่าการส่งมอบแบบผ่านสื่อสังคมออนไลน์

Hakli (2559, น. 38-42, 46) นำเสนอการประยุกต์ใช้ระบบกระบวนการการปรับใช้และส่งมอบ เพื่อแก้ปัญหาด้านระยะเวลาที่ล่าช้าในการพัฒนาซอฟต์แวร์กับองค์กร โดยอาศัยชุดเฟรมเวิร์คการทำงานการปรับใช้และส่งมอบแบบอัตโนมัติจากระบบคลาวด์ของ AWS เพื่อประยุกต์ใช้ในกระบวนการ Build การทดสอบและการปรับใช้งาน ส่งมอบแก่ผู้ใช้งาน โดยจากการวิจัยเป็นการทดลองกับภายในทีมงานวิจัยซึ่งแยกกระบวนการแต่ละกระบวนการ ตั้งแต่ขั้นตอนการออกแบบจนถึงขั้นตอนการส่งมอบและนำไปใช้งานอย่างชัดเจน ซึ่งสามารถนำไปใช้งานได้อย่างมีประสิทธิภาพ แต่ก็ทำให้เกิดค่าใช้จ่ายที่สูงเช่นกัน

Ton (2560, น. 5, 10) นำเสนอการพัฒนาระบบการทดสอบอัตโนมัติแบบยูนิต เพื่อตรวจสอบการทำงานของระบบที่ใช้ส่งมอบและนำไปใช้งานจริงผ่านกระบวนการอัตโนมัติด้วย Jenkins จากการอาศัยกระบวนบูรณาการ การปรับใช้และส่งมอบแบบต่อเนื่องในการส่งมอบแก่ผู้ใช้งานที่ไม่สามารถทำการทดสอบได้หลังผ่านกระบวนการส่งมอบไปแล้ว โดยประยุกต์ใช้ Spock Testing ทำการพัฒนา Unit Testing แบบอัตโนมัติและนำมาปรับใช้กับกระบวนการการทำงานของ Jenkins เพื่อให้ทำการทดสอบแบบอัตโนมัติก่อนทำการส่งมอบแก่ผู้ใช้งาน ทำให้เกิดความมั่นใจและความเชื่อมั่นในประสิทธิภาพการทำงานของซอฟต์แวร์ที่ส่งมอบให้แก่ผู้ใช้งาน

Juopperi (2560, น. 11-13) นำเสนอการเปรียบเทียบการทำงานระบบการส่งมอบการนำไปใช้แบบอัตโนมัติและการทำงานบนเครื่องใช้งานจริงในรูปแบบอื่น โดยเปรียบเทียบความแตกต่างหลังการพัฒนาระบบ ChatOps และนำกระบวนการอัตโนมัติด้วย Ansible แทนการปรับเปลี่ยนเครื่องแม่ข่ายที่รองรับการส่งมอบการนำไปใช้งานแบบอัตโนมัติ ซึ่งการทำงานแบบอัตโนมัติที่ปรับปรุงขึ้นสามารถช่วยให้การทำงานในธุรกิจขนาดเล็กและไม่มีบุคลากรในการสนับสนุนการส่งมอบการนำไปใช้สามารถทำงานต่อไปได้อย่างอัตโนมัติและทำให้สามารถควบคุมเวลาการทำงานได้

Arachchi and Perera (2561, น. 4-11) นำเสนอการประยุกต์กระบวนการบริหารจัดการแบบ Agile Methodology มาช่วยสนับสนุนการทำงานแบบบูรณาการ และการปรับใช้แบบต่อเนื่องให้เกิดประสิทธิภาพ โดยอาศัยการพัฒนาชุดคำสั่งไปป์ไลน์เพื่อแก้ปัญหาการส่งมอบการนำไปใช้ที่ล้มเหลว เนื่องจากปัญหาด้านระยะเวลาทำให้เกิดการล่าช้าในกระบวนการพัฒนาและทดสอบระบบแบบเดิม โดยการพัฒนาชุดคำสั่งไปป์ไบน์ขึ้น ทำงานร่วมกับ Jenkins, Ansible และ Git เพื่อสร้างกระบวนการทำงานแบบอัตโนมัติ และมีเครื่อง Nexus ในการเก็บ Release การทำพัฒนาซอฟตแวร์ ซึ่งการประยุกต์ใช้การทำงานแบบอัตโนมัตินี้สามารถช่วยลดความเสี่ยงในการนำไปใช้งานได้อย่างมีนัยสำคัญ

Faustino (2561, น. 33-54) นำเสนอการศึกษากรรมวิธีในการแก้ไขปัญหาที่เกิดขึ้นจากการนำเอาขั้นตอนการพัฒนาและปฏิบัติการเข้ามาใช้ โดยมีจุดมุ่งหมายเพื่อค้นพบปัญหาที่แปลกใหม่ที่เกิดขึ้น พร้อมกับแนวทางแก้ไขปัญหา ซึ่งทำให้ผู้ที่เคยพบปัญหาเหล่านี้เกิดข้อได้เปรียบในเชิงวิชาการที่สามารถทำให้ตนเองสามารถแก้ไขปัญหาและสาเหตุที่ทำให้เกิดปัญหาได้อย่างรวดเร็ว เพื่อเป็นการนำเอาเทคนิคและวิธีการมาสร้างเป็นแหล่งเรียนรู้และการกำหนดทิศทางมาตรฐานแก่ ITSM และ ITIL ด้วยกรรมวิธีการสังเกต วิเคราะห์เอกสาร และสัมภาษณ์แบบกึ่งโครงสร้าง ซึ่งพบว่า แนวทางการปฏิบัติและขั้นตอนที่ใช้แก้ไขปัญหาที่ได้ทำการวิจัย มีความสัมพันธ์และสามารถนำไปใช้ในการแก้ปัญหาได้

Ivanov (2561, น. 44-53) นำเสนอการพัฒนาการประมวลผลแบบไร้เครื่องแม่ข่ายให้เกิดรูปแบบการประมวลผลแบบคลาวด์ โดยอาศัยกระบวนการทำงานที่ประมวลผลแบบคอนเทรนเนอร์ มาใช้ในการจัดการทรัพยากร เพื่อพัฒนาเป็นแอปพลิเคชันแบบ Function as a Service (FaaS) โดยอาศัยกระบวนการ DevOps เป็นตัวขับเคลื่อนการพัฒนาแนวทางการออกแบบสถาปัตยกรรมซอฟต์แวร์และงานพัฒนาซอฟต์แวร์ด้วยคำสั่งไปป์ไลน์สำหรับใช้งานภายในองค์กร ซึ่งผลการพัฒนาโมเดลไปป์ไลน์ดังกล่าวทำให้เกิดการพัฒนาโดยไร้เครื่องแม่ข่ายในงานพัฒนาแอปพลิเคชันใหม่เพิ่มขึ้น และเป็นการนำกระบวนการบูรณาการและการนำไปใช้งานแบบต่อเนื่องเข้ามาใช้เพิ่มความรวดเร็วในการนำไปใช้งาน

Paloposki (2561, น. 28-31, 37) นำเสนอการนำระบบอัตโนมัติมาช่วยปรับกระบวนการการพัฒนาและทดสอบเพื่อยอมรับระบบโดยใช้ Ansible เป็นตัวปรับกระบวนการปรับใช้งานแบบอัตโนมัติในการเปิดตัวซอฟต์แวร์ใหม่ จากปัญหาเดิมที่มีการพัฒนาซอฟต์แวร์และเปิดตัวได้น้อยครั้ง จึงนำกระบวนการอัตโนมัติเข้ามาปรับใช้ให้เกิดความถี่ในการปรับใช้งานและพัฒนาซอฟต์แวร์ใหม่ให้เกิดมากขึ้น ซึ่งจากผลการวิจัยพบว่าการนำเอากระบวนการอัตโนมัติเข้ามาปรับใช้งาน สามารถทำให้เกิดการพัฒนาซอฟต์แวร์ใหม่เพิ่มขึ้นตามสมมติฐานที่ได้คาดการณ์ไว้ และยังสามารถทำให้เกิดการปรับและนำไปใช้เพิ่มขึ้นในอุตสาหกรรมการทดสอบและปรับปรุงคุณภาพซอฟต์แวร์ด้วย

Nair (2562, น. 34-69) นำเสนอการพัฒนาระบบประมวลผลชุดคำสั่ง ไฟล์ รวมถึงไลบลารี่ต่างๆ ที่ใช้ในการบีบอัดการส่งมอบเพื่อนำไปใช้งานจากการพัฒนาซอฟต์แวร์ด้วยการประมวลผลแบบอัตโนมัติในการคอมไพล์และดำเนินการ โดยเน้นการเพิ่มความรวดเร็วในการทำงานด้วยโครงสร้างการพัฒนาแบบจำลองโมเดลที่เป็นเครื่องมือสนับสนุนการพัฒนาซอฟต์แวร์ โดยสามารถปรับปรุงประสิทธิภาพของแบบจำลองโมเดลได้แบบอัตโนมัติทันที และวัดผลจากระยะเวลารวมในการประมวลผลของแต่ละองค์ประกอบแบบจำลอง

Srivastava (2562, น. 37-53) นำเสนอการพัฒนาระบบการทำงานแบบต่อเนื่องจากการตรวจจับภาพโดยอาศัย Deep Learning ในการเรียนรู้และประมวลผลบนระบบคลาวด์ เพื่อหาข้อผิดพลาดในชิ้นส่วนที่ใช้ในการผลิตรถยนต์ด้วยภาพ โดยใช้ภาษา Python ร่วมกับไลบลารี่ BOTO3 ในการประมวลผล เรียนรู้และตรวจจับภาพชิ้นส่วนรถยนต์ เพื่อลดปริมาณงานและข้อผิดพลาดที่อาจเกิดขึ้นได้จากมนุษย์ด้วยระบบอัตโนมัติ

ซึ่งจากการศึกษางานวิจัยที่เกี่ยวข้องทั้งหมดพบว่าเครื่องมือในแต่ละงานวิจัยที่เกี่ยวข้องส่วนใหญ่มีการนำเอาระบบเซิร์ฟเวอร์อัตโนมัติเข้ามาใช้ในการช่วยพัฒนาเพิ่มประสิทธิภาพในการทำงาน แต่มักใช้งานเป็นระบบเซิร์ฟเวอร์อัตโนมัติแบบสำเร็จรูปและนำเอาส่วนเสริมที่เป็นการทำงานสำคัญที่มีให้ใช้งานอยู่แล้วมาใช้ในการพัฒนาและดำเนินการวิจัย อันเป็นผลให้เกิดข้อจำกัดในหลายด้าน เช่น การปรับเปลี่ยนรูปแบบการทำงานตามบริบทที่จำเป็นของหน่วยงาน ความเหมาะสมกับการทำงานในบริบทองค์กรขนาดใหญ่โดยพิจารณาจากหน่วยงานหรือขอบเขตประชากรและกลุ่มตัวอย่างของงานวิจัยที่เกี่ยวข้อง รวมถึงการลงทุนจากการสมัครหรือใช้งานระบบเซิร์ฟเวอร์อัตโนมัติแบบสำเร็จรูปซึ่งมีค่าใช้จ่ายสูงในการขอรับบริการเพื่อการประมวลผลบนคลาวด์ ยกตัวอย่างเช่น Amazon Web Services หรือ Microsoft Azure เป็นต้น อันส่งผลให้หน่วยงานขนาดเล็กหรือหน่วยงานที่มี

ข้อจำกัดด้านการลงทุนมักไม่ให้ความสนใจใช้งานเว้นแต่มีความจำเป็น ทำให้ในหน่วยงานที่มีอุปกรณ์เพื่อรองรับการทำงานในการพัฒนาระบบเซิร์ฟเวอร์อัตโนมัติของตนเองอยู่แล้วนั้น ไม่ได้ถูกนำเอาอุปกรณ์ต่างๆไปใช้งานอย่างเต็มที่จากการลงทุน ทั้งนี้ยังมีข้อจำกัดในการพัฒนาระบบเซิร์ฟเวอร์อัตโนมัติโดยมากจากการศึกษางานวิจัยที่เกี่ยวข้องในเรื่องของการทำงานบนระบบปฏิบัติการ Open Source อาทิ Redhat, UNIX รวมถึง Linux เป็นต้น

จากข้อจำกัดของงานวิจัยที่เกี่ยวข้องที่ได้กล่าวนั้น ผู้วิจัยจึงนำโอกาสจากงานวิจัยที่เกี่ยวข้องในการพัฒนาระบบเซิร์ฟเวอร์อัตโนมัติโดยอาศัยการพัฒนากระบวนการทำงานตามบริบทการทำงานด้วย Open Source บนชุดคำสั่งการทำงาน Pipeline ของ Jenkins รวมเข้ากับการทำงานบนระบบปฏิบัติการ Windows Server ซึ่งหลายหน่วยงานยังคงใช้งานอยู่แต่มักต้องใช้กับ Microsoft Azure บนการประมวลผลแบบคลาวด์เท่านั้น ดังนั้นงานวิจัยนี้จะเป็นโอกาสที่จะนำเอาขั้นตอนและกระบวนการพัฒนาจากอุปกรณ์ที่มีอยู่ไปใช้งานให้เกิดประสิทธิภาพสูงสุดและให้เกิดผลลัพธ์ตามวัตถุประสงค์ของการวิจัยด้วยเครื่องมือวัดประสิทธิภาพในด้านการจัดการเวลาในการประมวลผล เพื่อเปรียบเทียบการทำงานของ 2 กระบวนการ ตามที่งานวิจัยที่เกี่ยวข้องร้อยละ 60 ของงานวิจัยที่เกี่ยวข้องที่ได้ทำการศึกษาได้เลือกนำมาใช้ในการวัดประสิทธิภาพการทำงานของระบบเซิร์ฟเวอร์อัตโนมัติ โดยมีการเรียกใช้ในชื่อเรียกที่แตกต่างกันออกไปตามการตั้งขึ้นเพื่อให้สอดคล้องกับงานวิจัยของตนเอง แต่สามารถสรุปได้ว่าเป็นเครื่องมือที่ใช้วัดผลด้านเวลาการทำงานหรือการประมวลผลของงานวิจัยที่พัฒนาขึ้นสำหรับนำไปใช้ในการเปรียบเทียบผลด้านระยะเวลาตามวัตถุประสงค์

บทที่ 3
วิธีดำเนินการวิจัย

วิทยานิพนธ์นี้ผู้วิจัยได้ดำเนินการวิจัยให้ครอบคลุม (1) ประชากรและกลุ่มตัวอย่าง (2) เครื่องมือการวิจัย (3) วิธีการรวบรวมข้อมูล (4) วิธีการวิเคราะห์และนำเสนอข้อมูล โดยมีรายละเอียดวิธีการดำเนินการวิจัยดังต่อไปนี้

1. ประชากรและกลุ่มตัวอย่าง

ได้เลือกประชากรมาจากการสำรวจปริมาณจำนวนงานของบุคลากรที่ประกอบอาชีพด้านเทคโนโลยีสารสนเทศ ที่มีส่วนเกี่ยวข้องกับกระบวนการพัฒนาระบบ ทดสอบระบบ และการติดตั้งระบบเพื่อใช้งานในทางธุรกิจ ภายในอาคารพหลโยธินเพลส ชั้น 23 เขตพญาไท กรุงเทพมหานคร ในปีพุทธศักราช 2563 จำนวน 86 คน โดยใช้กำหนดเป็นปริมาณจำนวนงานตามจำนวนประชากรได้จำนวน 86 งาน ซึ่งสำหรับการทำการวิจัยในครั้งนี้ใช้วิธีการวิจัยในเชิงการทดลองและวิเคราะห์ข้อมูลจากการเปรียบเทียบผลการทดสอบสำหรับวัดผลความสำเร็จจากกลุ่มตัวอย่างด้วยวิธีการสุ่มอย่างง่าย (Simple Random Sampling) จากการเปิดตารางเครจซี่และมอร์แกน (Krejcie & Morgan) ที่สามารถนำไปใช้ในการกำหนดกลุ่มตัวอย่างได้เป็นจำนวนทั้งสิ้น 75 ตัวอย่างงาน แต่ด้วยจำนวนของประชากรที่มีขนาดไม่มากนัก ผู้วิจัยจึงทำการกำหนดกลุ่มตัวอย่างด้วยจำนวนประชากรทั้งหมดโดยคิดเป็นปริมาณงานทั้งสิ้น 86 งาน ตามเงื่อนไขการกำหนดงานที่นำมาใช้ในการทดสอบประสิทธิภาพจากงานที่เป็นกรณีเร่งด่วนตามข้อกำหนดการตกลงระดับบริการที่ต้องได้รับการแก้ไขแบบเร่งด่วนภายในกำหนดการส่งมอบงานที่ผ่านการพัฒนาหรือแก้ไขไม่เกิน 3 วัน ซึ่งสามารถแยกตามขนาดของงานได้ ทั้งสิ้น 3 ขนาด ตามการระบุภายในข้อตกลงระดับบริการที่หน่วยงานได้กำหนดและได้ระบุขึ้นภายใต้หลักทฤษฎีของ COCOMO ประกอบด้วยขนาด S, M และ L หรือขนาดเล็ก ขนาดกลาง และขนาดใหญ่ตามลำดับ ซึ่งหน่วยงานเป็นผู้กำหนดขึ้นจากการคำนวณตามหลักการทฤษฎี COCOMO ที่อาศัยการวัดจากระยะเวลาในการพัฒนา ปริมาณทีมหรือจำนวนคนที่ใช้ในการพัฒนา กลุ่มเป้าหมายที่ใช้งาน จำนวนบรรทัดที่ใช้ในการพัฒนาโปรแกรม (Line of Code : LOC) รวมถึงขนาดของโครงสร้างและ

สถาปัตยกรรมซอฟต์แวร์ (Function Point : FP) เพื่อเข้าสู่กระบวนการประเมินขนาดของระบบเป็นมูลค่าต้นทุนด้วยสมการก่อนนำมาใช้เปรียบเทียบเชิงปริมาณกับซอฟต์แวร์อื่น และกำหนดเป็นขนาดของซอฟต์แวร์ขึ้นตามหลักการข้อกำหนดของหน่วยงานที่ได้กำหนดไว้ดังตัวอย่างตารางที่ 3.1

ตารางที่ 3.1 ตัวอย่างการแบ่งกลุ่มขนาดของซอฟต์แวร์ระบบ

ขนาด	**ซอฟต์แวร์**	**กลุ่มเป้าหมาย**
เล็ก (S)	ระบบจัดการและการส่งต่อปัญหา ระบบติดตามโครงการและงาน ระบบบันทึกการปฏิบัติงานรายวัน ระบบดูแลและติดตามเรื่องร้องเรียน	ภายในองค์กร
กลาง (M)	ระบบจัดการบัญชีและการเงิน ระบบจัดซื้อและขนส่ง	ตัวแทนองค์กรและสาขา
ใหญ่ (L)	ระบบจองซื้อและปรับเปลี่ยนบริการ ระบบข้อมูลลูกค้าสัมพันธ์ ระบบบริการข้อมูลด้วยตนเอง	ลูกค้าทั่วประเทศ

และด้วยสถานการณ์การแพร่ระบาดของไวรัสโคโรน่า 2019 (COVID-19) ที่กลุ่มประชากรและกลุ่มตัวอย่างทำงานผ่านการควบคุมเครื่องระยะไกล ทำให้ปริมาณงานสำหรับทดสอบมีจำกัดจึงเลือกนำงานทดสอบทั้งหมดเกิดขึ้นจากข้อมูลงานทดสอบที่ใช้งานจริง โดยไม่ได้อ้างอิงหรือแบ่งจากงานสำหรับการทดสอบงานใดงานหนึ่งเท่านั้น ทำให้สัดส่วนของการเลือกงานทดสอบตามขนาดซอฟต์แวร์ที่กล่าวไว้ในตารางที่ 3.1 เป็นสัดส่วนที่เกิดขึ้นจากปริมาณการทำงานจริงตามช่วงเวลาที่ทำการทดสอบและขอบเขตงานเร่งด่วนที่กำหนดไว้ข้างต้น

2. เครื่องมือการวิจัย

เครื่องมือที่ใช้ในการวิจัยในครั้งนี้ ได้แก่ (1) ชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps (2) โปรแกรมเสริมความสามารถ Jenkins Pipeline สำหรับพัฒนาชุดคำสั่งโปรแกรมเพื่อวัดประสิทธิภาพการทำงานของออโตเมชันเซิร์ฟเวอร์ (3) แบบบันทึกผลปริมาณจำนวนงานของกลุ่มตัวอย่างจากการทำงานด้วยวิธีการแบบแมนวล (Manual) และออโตเมชันตามแนวคิด CI/CD ของกระบวนการ DevOps และ (4) แบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน Service Level Agreements ทางธุรกิจ ซึ่งมีรายละเอียดการพัฒนาเครื่องมือวิจัยแต่ละเครื่องมือดังนี้

2.1 ชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps

ผู้วิจัยได้พัฒนาออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อจำลองและทดแทนการทำงานแบบวิธีการมือ (Manual) ให้สามารถทำงานได้อย่างอัตโนมัติตามการตั้งค่าและการโปรแกรมการทำงานไว้ ซึ่งมีขั้นตอนในการศึกษา พัฒนา และทดสอบการใช้งานดังนี้

2.1.1 ขั้นตอนการศึกษา

1) ศึกษาทฤษฎีและวิธีการพัฒนาออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps

2) ศึกษาเครื่องมือที่ใช้ในการพัฒนาออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ได้แก่

2.1) เครื่องมือ Source Control ด้วย GitLab

2.2) เครื่องมือควบคุมการทำงานออโตเมชันด้วย Jenkins และภาษา Groovy

2.3) เครื่องมือจำลอง Server ด้วย IIS

2.4) เครื่องมือสำหรับการ Build Source Code ด้วย MS-Build

2.5) เครื่องมือสำหรับพัฒนาซอฟต์แวร์

2.5.1) Microsoft Visual Studio 2017

2.5.2) ภาษาที่ใช้พัฒนา ASP.NET (C#.NET) MVC4

2.5.3) .NET Framework 4.0 ขึ้นไป

2.6) Microsoft Windows Server 2012 ขึ้นไป

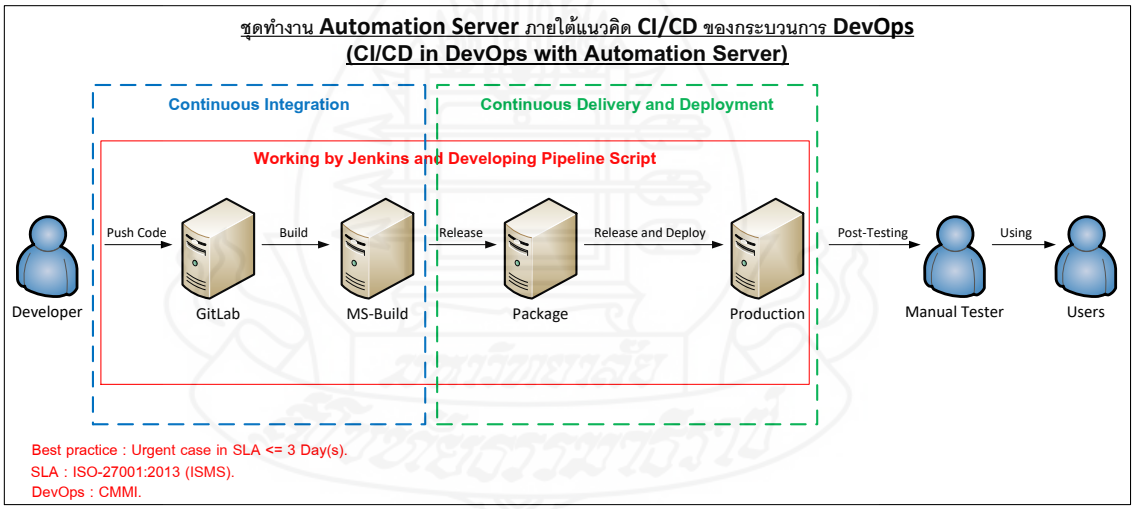
3) ศึกษาข้อกำหนดระดับการบริการทางธุรกิจของหน่วยงานที่ถูกกำหนดขึ้นเพื่อให้เป็นไปตามมาตรฐาน ISO-27001 : 2013 หมวดระบบบริหารจัดการความมั่นคงปลอดภัยสารสนเทศ (ISMS) ที่หน่วยงานได้ผ่านการรับรองและได้ถูกกำหนดไว้แล้วในรูปแบบของข้อตกลงระดับบริการทางธุรกิจที่มีรายละเอียดการชี้แจงเกี่ยวกับวิธีการแก้ไขปัญหา ระยะเวลา ผู้รับผิดชอบ และตัวชี้วัดต่างๆ

2.1.2 ขั้นตอนการพัฒนา

1) ติดตั้งเครื่องมือที่จำเป็นต่อการพัฒนาชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ตามที่ได้ทำการศึกษา

2) พัฒนาชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ตามที่ได้ทำการศึกษาด้วยการติดตั้งและโปรแกรมคำสั่งเพื่อให้ทำงานตามกระบวนการขั้นตอนวิธีอัตโนมัติที่ถูกต้อง โดยการทำงานของชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เป็นไปตามกระบวนการทำงานโดยเริ่มตั้งแต่ผู้พัฒนาซอฟต์แวร์ผ่านการยืนยันการจัดเก็บชุดคำสั่งที่ได้รับการแก้ไข (Commit Source Code) ขึ้น GitLab และดำเนินการด้วยแนวคิด CI/CD แต่การทำงานจะดำเนินการโดยข้ามขั้นตอนของการทดสอบที่ต้องอาศัยการทดสอบอัตโนมัติ (Automation Testing) เข้ามาทำงาน ที่สามารถทำงานได้เพียง 1 ซอฟต์แวร์ต่อ 1 การทดสอบเท่านั้น ในขั้นตอนนี้จึงมีการปรับเปลี่ยนเป็นการทดสอบด้วยวิธีการทดสอบแบบแมนวล (Manual Testing) แทน ดังนั้นตามจุดประสงค์ของการวิจัยที่มีจุดประสงค์สำคัญที่การวัดประสิทธิภาพการทำงานเพื่อตอบสนองข้อตกลงระดับบริการจึงดำเนินการตามกระบวนการตามภาพที่ 3.1 ซึ่งอธิบายกระบวนการทำงานของออโตเมชันเซิร์ฟเวอร์จากงานทดสอบที่ถูกกำหนดขึ้นตาม SLA ระดับเร่งด่วน (Urgent Case) เนื่องจากเป็นงานที่ต้องได้รับการพัฒนาหรือแก้ไขซอฟต์แวร์ให้เสร็จสิ้นภายในระยะเวลาไม่เกิน 3 วัน ตามขอบเขตการวิจัยที่กำหนดเพื่อไม่ให้เกิดระยะเวลาที่ใช้ในการทดสอบที่นานเกินไป อีกทั้งเป็นงานที่เป็นปัจจัยสำคัญในการเกิดโอกาสอัตราความสำเร็จที่มีความสำเร็จตาม SLA ในเกณฑ์ที่ต่ำหรือน้อยมาก จากการใช้ระยะเวลาในการพัฒนาหรือแก้ไขซอฟต์แวร์ที่จำกัด ซึ่งนักพัฒนาจะทำการพัฒนาหรือแก้ไขซอฟต์แวร์และทำการทดสอบซอฟต์แวร์เบื้องต้นด้วยการทดสอบระบบแบบ

หน่วยทดสอบ (Unit Testing) โดยเมื่อผ่านการทดสอบและพร้อมใช้งานนักพัฒนาจะดำเนินการส่งคำสั่ง Push Code ขึ้นสู่ Source Control บน GitLab และส่งคำสั่งการทำงานไปยังออโตเมชันเซิร์ฟเวอร์ที่พัฒนาขึ้นด้วย Jenkins และ Pipeline Script รวมถึงคำสั่ง Command Line ผ่านการประมวลผลแบบชั่วคราว (Batch Processing) ด้วยวิธีการ Trigger จาก GitLab ทันทีที่นักพัฒนาได้ Push Code เสร็จสมบูรณ์ กระบวนการทำงานที่กำหนดขึ้นด้วยคำสั่ง Pipeline Script จะทำการส่งคำสั่งการสร้าง (Build) ไปยังตัวตัวควบคุมการสร้าง MS-Build เพื่อให้ทำการดึงข้อมูล Source Code จาก GitLab เข้าสู่ขั้นตอน Build เป็นภาษาที่คอมพิวเตอร์สามารถเข้าใจได้ เพื่อส่งต่อไปยังกระบวนการสร้างเป็นเวอร์ชันกลุ่มโปรแกรมสำหรับการนำไปใช้เผยแพร่ (Release) ก่อนทำกระบวนการจัดกลุ่มโปรแกรมสำเร็จ (Package) เป็นชุดโปรแกรมที่สามารถนำไปใช้งานได้เผยแพร่และถูกนำไปติดตั้งใช้งาน (Deploy) บนพื้นที่สำหรับทำงานที่กำหนดไว้ (Application Path) บนเครื่องแม่ข่ายสำหรับการใช้งานจริง (Production) ซึ่งผู้ทดสอบระบบจะดำเนินการทดสอบระบบหลังการนำไปใช้งานจริงบนเครื่องแม่ข่าย (Post-Test) เพื่อยืนยันผลการทำงานที่สมบูรณ์แก่ผู้ใช้งานต่อไป ดังภาพที่ 3.1



ภาพที่ 3.1 กระบวนการทำงานตามแนวคิดการวิจัย

2.1.3 ขั้นตอนการทดสอบและใช้งาน

ขั้นตอนการทดสอบและใช้งานชุดทำงานออโตเมชันเซิร์ฟเวอร์ตามแนวคิดการวิจัย ประกอบด้วย

1) ใช้ซอฟต์แวร์ที่ถูกพัฒนาให้ทำงานผ่านเว็บเบราส์เซอร์ในรูปแบบของการทำงานเว็บแอปพลิเคชันหรือเว็บเซอร์วิสด้วย ASP.NET ภายใต้พื้นฐานการทำงาน .NET Framework 4.0 ขึ้นไป เพื่อทำการพัฒนาตามขั้นตอนการพัฒนา (Code)

2) ทำการทดสอบแบบหน่วยทดสอบ (Unit Test) เพื่อทดสอบการทำงานแบบแมนวล (Manual Testing) ของเว็บแอปพลิเคชันที่พัฒนาขึ้นว่าสามารถใช้งานได้อย่างเป็นปกติ

3) จัดเก็บชุดคำสั่งโปรแกรมที่พัฒนาและพร้อมใช้งานสู่แหล่งจัดเก็บชุดคำสั่งด้วย GitLab ภายใต้แนวคิดของกระบวนการ DevOps ขั้นตอนการพัฒนา (Code)

4) GitLab ทำการจัดเก็บข้อมูลชุดคำสั่งที่ถูกแก้ไขและทำการ Trigger เพื่อส่งต่อการทำงานไปยัง Jenkins เพื่อสั่งงานให้ Jenkins Pipeline เริ่มทำงานตามคำสั่งที่ถูกกำหนดไว้

5) ตรวจสอบการทำงานของ Jenkins ในการดำเนินการต่างๆ ภายใต้แนวคิดของกระบวนการ DevOps ดังนี้

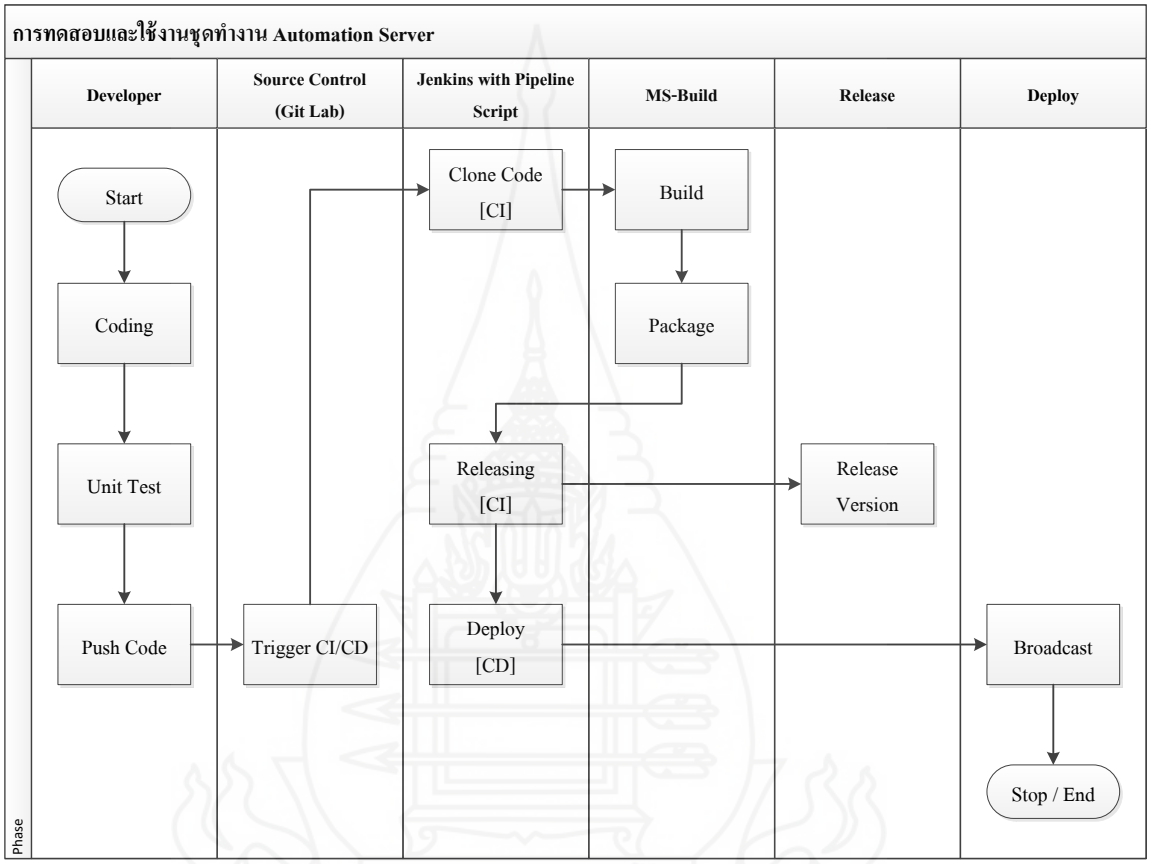
5.1) ขั้นตอนการ Build ด้วยการสั่งการของ Jenkins เพื่อทำการดึง Source Code จากขั้นตอนการพัฒนา (Code) บน GitLab ด้วยคำสั่ง Clone ไปยังพื้นที่ทำงานที่กำหนด

5.2) MS-Build ทำงานในการ Build Source Code จากพื้นที่ทำงานที่กำหนดและเกิดการจัดกลุ่มโปรแกรมสำเร็จ (Package) เป็นชุดคำสั่งพร้อมใช้งานในรูปแบบของเผยแพร่ (Release)

5.3) ขั้นตอนการเผยแพร่ (Release) ในขั้นตอนนี้ Jenkins จะดำเนินการคัดลอกชุดคำสั่งที่พร้อมใช้งานในรูปแบบของการเผยแพร่เพื่อนำไปใช้งานต่อ

5.4) ขั้นตอนการติดตั้ง (Deploy) การทำงานนี้จะสั่งงานให้ Jenkins นำชุดคำสั่งจากการเผยแพร่ (Release) ที่พร้อมใช้งานไปทำการวางในส่วนของการนำออกไปใช้งานจริง เพื่อนำไปใช้งานในเครื่องแม่ข่ายที่ให้บริการอยู่ (Production) โดยจากการวิจัยครั้งนี้จะถูกนำไปวางในพื้นที่ส่วนของการบริหารจัดการข้อมูลอินเทอร์เน็ต (IIS) ภายในเครื่องแม่ข่ายที่ให้บริการที่ถูกสร้างเป็นกลุ่มเว็บไซต์สำหรับการให้บริการ (Site) และเปิดใช้งานจริง (Broadcast)

5.5) ทดสอบการใช้งานด้วยการเรียกให้เว็บแอปพลิเคชัน หรือ เว็บเซอร์วิสทำงาน โดยการทำงานจะต้องสามารถทำงานได้ตาม Unit Test ที่ถูกทำการทดสอบก่อนนำมาใช้งานจริง ซึ่งขั้นตอนนี้อยู่นอกเหนือขอบเขตของการวิจัยในครั้งนี้



ภาพที่ 3.2 ขั้นตอนการทดสอบและใช้งานชุดทำงานออโตเมชันเซิร์ฟเวอร์ตามแนวคิดการวิจัย

2.2 โปรแกรมเสริมความสามารถ Jenkins Pipeline สำหรับพัฒนาชุดคำสั่งโปรแกรมเพื่อวัดประสิทธิภาพการทำงานของออโตเมชันเซิร์ฟเวอร์

ผู้วิจัยได้ศึกษาเครื่องมือ และพัฒนาชุดคำสั่งสำหรับการวัดประสิทธิภาพการทำงานของออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อใช้ในการวัดประเมินประสิทธิภาพการทำงานของของออโตเมชันเซิร์ฟเวอร์ ให้เกิดความเชื่อมั่นและน่าเชื่อถือด้วย

โปรแกรมเสริมความสามารถสำเร็จรูปผ่าน Jenkins Pipeline เพื่อวัดประสิทธิภาพด้านกระบวนการ DevOps สำหรับรับรองการใช้งานให้สนับสนุน SLA ทางธุรกิจ ดังนี้

2.2.1 ศึกษาเครื่องมือและวิธีการพัฒนาเครื่องมือวัดประสิทธิภาพของออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps

2.2.2 เลือกเครื่องมือวัดประสิทธิภาพสำเร็จรูปที่สามารถทำงานร่วมกับชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ได้

2.2.3 ทดสอบและทดลองพัฒนาเครื่องมือวัดประสิทธิภาพด้วยชุดคำสั่งควบคุมการทำงานตามที่กำหนด

2.2.4 พัฒนาโปรแกรมเสริมความสามารถ Jenkins Pipeline สำหรับพัฒนาชุดคำสั่งโปรแกรมเพื่อวัดประสิทธิภาพการทำงานของออโตเมชันเซิร์ฟเวอร์

2.2.5 สรุปผลการวัดประสิทธิภาพการทำงานของออโตเมชันเซิร์ฟเวอร์โดยอาศัยการเปรียบเทียบตามเกณฑ์การวัดประสิทธิภาพ ดังนี้

1) ระยะเวลาเฉลี่ย (Average Stage Time) ในแต่ละพื้นที่ทำงาน (Stage) ของออโตเมชัน น้อยกว่าการทำงานแบบวิธีการมือ (Manual) ซึ่งใช้การคำนวณด้วย Jenkins Pipeline Plugin เป็นเครื่องมือช่วยในการคำนวณระยะเวลาเฉลี่ย โดยเป็นไปตามสถิติค่าเฉลี่ยรายพื้นที่ดังนี้

$$(\bar{X}) = \frac{\sum fx}{n} \quad (1)$$

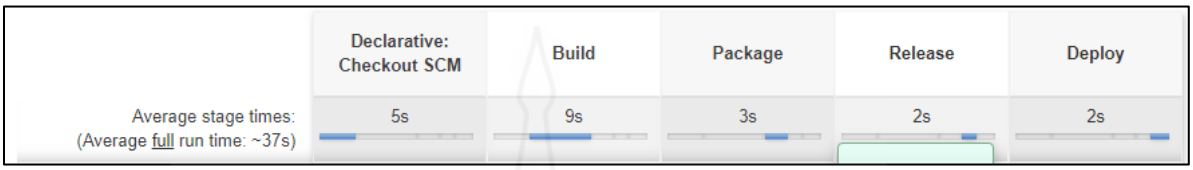
เมื่อ $(\bar{X})$ แทน ค่าเฉลี่ยที่ได้จากการคำนวณ

n แทน จำนวนปริมาณงานที่ใช้ในการทดสอบ

$\sum fx$ แทน เวลาทำงานรวมทั้งหมดของพื้นที่ทำงานหนึ่ง

2) ระยะเวลาเฉลี่ยรวมทั้งการทำงานโดยรวมทุกพื้นที่ทำงาน (Average Full Runtime) ในแต่ละการทำงานของออโตเมชันน้อยกว่าการทำงานแบบวิธีการมือ

(Manual) โดยอาศัยการรวบรวมระยะเวลาทำงานรวมทุกพื้นที่ทำงานด้วย Jenkins Pipeline Plugin จากการนำระยะเวลารวมในแต่ละพื้นที่ของแต่ละงานนำมาคำนวณเป็นค่าเฉลี่ยเวลา ดังภาพที่ 3.3



ภาพที่ 3.3 การแสดงระยะเวลาเฉลี่ยรวมแต่ละพื้นที่และระยะเวลาเฉลี่ยรวมทุกพื้นที่ทำงานด้วย Jenkin Pipeline Plugin

3) ระยะเวลา Average Full Runtime ในแต่ละการทำงานทดสอบไม่เกินเงื่อนไขระยะเวลาของ SLA ตามที่ถูกกำหนดไว้

2.3 แบบบันทึกผลปริมาณจำนวนงานของกลุ่มตัวอย่างจากการทำงานด้วยวิธีการแบบแมนวล (Manual) และออโตเมชันตามแนวคิด CI/CD ของกระบวนการ DevOps

ผู้วิจัยได้ศึกษาแนวคิด CI/CD ของกระบวนการ DevOps เพื่อนำไปใช้ในการพัฒนา และจัดทำแบบบันทึกผลปริมาณจำนวนงานของกลุ่มตัวอย่างจากการทำงานด้วยวิธีการแบบแมนวล (Manual) และออโตเมชันตามแนวคิด CI/CD ของกระบวนการ DevOps สำหรับนำไปใช้ในการวิเคราะห์ข้อมูลร่วมกับโปรแกรมเสริมความสามารถ Jenkins Pipeline สำหรับพัฒนาชุดคำสั่งโปรแกรมเพื่อวัดประสิทธิภาพการทำงานของออโตเมชันเซิร์ฟเวอร์โดยมีขั้นตอนดังนี้

2.3.1 ศึกษาทฤษฎีและแนวคิด CI/CD ของกระบวนการ DevOps

2.3.2 พัฒนาฉบับร่างแบบบันทึกผลปริมาณจำนวนงานของกลุ่มตัวอย่างจากการทำงานด้วยวิธีการแบบแมนวล (Manual) และออโตเมชันตามแนวคิด CI/CD ของกระบวนการ DevOps

2.3.3 จัดทำแบบบันทึกผลปริมาณจำนวนงานของกลุ่มตัวอย่างจากการทำงานด้วยวิธีการแบบแมนวล (Manual) และ ออโตเมชัน ตามแนวคิด CI/CD ของกระบวนการ DevOps ฉบับสมบูรณ์

2.4 แบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ

ผู้วิจัยได้ศึกษา พัฒนา และจัดทำแบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps เพื่อเพิ่มประสิทธิภาพข้อตกลงระดับการบริการ สำหรับให้กลุ่มตัวอย่างทำการประเมินหลังการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ที่ใช้ในการวิจัยการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps โดยมีขั้นตอน ดังนี้

2.4.1 ศึกษาทฤษฎีและวิธีการพัฒนาออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps

2.4.2 ศึกษาทฤษฎีและและแนวคิดเพื่อรองรับการสนับสนุน Service Level Agreements ทางธุรกิจ

2.4.3 สรุปทฤษฎีและแนวคิดเพื่อจัดทำฉบับร่างแบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน Service Level Agreements ทางธุรกิจ จำนวน 3 ตอน ได้แก่

1) ตอนที่ 1 ข้อมูลทั่วไปสำหรับผู้ตอบแบบประเมินการใช้งาน ซึ่งจะเป็นคำถามปลายปิดเกี่ยวกับการสอบถามสถานภาพ ช่วงอายุ เพศ ระดับการศึกษา และตำแหน่งงานของผู้ตอบแบบประเมินการใช้งาน โดยต้องทำการตอบคำถามทุกข้อ

2) ตอนที่ 2 แบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ ซึ่งเป็นคำถามปลายปิด ซึ่งต้องทำการตอบแบบประเมินการใช้งานทุกข้อ โดยแบ่งเป็น 3 ด้าน ดังนี้

2.1) การใช้งานและประสิทธิภาพ

2.2) การลดระยะเวลาและขั้นตอนการทำงานจากการใช้งาน

2.3) การทำงานตามเป้าหมายของ SLA

ทั้งนี้การตอบคำถามจะแบ่งระดับคะแนนออกเป็น 5 ระดับ ดังนี้

คะแนน 5 หมายถึง ระดับมากที่สุด

คะแนน 4 หมายถึง ระดับมาก

คะแนน 3 หมายถึง ระดับปานกลาง

คะแนน 2 หมายถึง ระดับน้อย

คะแนน 1 หมายถึง ระดับน้อยที่สุด

3) ตอนที่ 3 ข้อเสนอแนะ หรือ ความคิดเห็น เป็นคำถามปลายเปิด เพื่อให้กลุ่มตัวอย่างสามารถเสนอความคิดเห็นและข้อเสนอแนะต่างๆที่เกี่ยวข้องกับการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ ซึ่งกลุ่มตัวอย่างจะเสนอแนะความคิดเห็นหรือไม่ก็ได้

2.4.4 จัดทำแบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ ฉบับสมบูรณ์บน Google Form สำหรับให้กลุ่มตัวอย่างใช้ทำการประเมินการใช้งานแต่ละตอนผ่านเว็บเบราว์เซอร์และอินเทอร์เน็ต เพื่อความสะดวกในการเก็บรวบรวมข้อมูลและนำข้อมูลไปใช้ในการวิเคราะห์ผล

3. วิธีการรวบรวมข้อมูล

สำหรับวิธีการรวบรวมข้อมูลในการวิจัยนี้ มีวิธีการรวบรวมข้อมูล แบ่งออกเป็น 2 ขั้นตอน โดยมีรายละเอียดขั้นตอนการรวบรวมข้อมูลดังนี้

3.1 ขั้นตอนการเก็บรวบรวมข้อมูลการประเมินประสิทธิภาพการใช้งานของออโตเมชันเซิร์ฟเวอร์

3.1.1 ทำการเก็บข้อมูลกระบวนการทำงานแบบวิธีมือ (Manual) ของกลุ่มตัวอย่างด้วยระยะเวลาที่เกิดขึ้นบนซอฟตแวร์เว็บแอปพลิเคชัน สำหรับใช้ติดตามการทำงานตามกระบวนการ Deployment ระบบของหน่วยงาน โดยขั้นตอนนี้จะทำคู่ขนานไปกับขั้นตอนถัดไป

3.1.2 เก็บข้อมูลของกลุ่มตัวอย่างจากการใช้งานชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps จากโปรแกรมเสริมความสามารถ Jenkins Pipeline สำหรับวัดประสิทธิภาพการทำงานของออโตเมชันเซิร์ฟเวอร์โดยใช้งานทดสอบที่เป็นงานเดียวกันกับขั้นตอนก่อนหน้านี้

3.1.3 จัดเรียงและสรุปผลการใช้งานชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps และผลการทำงานตามกระบวนการทำงานแบบวิธีการมือ (Manual) ที่เกิดขึ้นบนซอฟต์แวร์เว็บแอปพลิเคชันสำหรับใช้ติดตามการทำงานตามกระบวนการ ติดตั้ง (Deployment) ระบบของหน่วยงาน ลงบนแบบบันทึกผลปริมาณจำนวนงานของกลุ่มตัวอย่างจากการทำงานด้วยวิธีการแบบแมนวล (Manual) และออโตเมชันตามแนวคิด CI/CD ของกระบวนการ DevOps

3.1.4 เปรียบเทียบ วิเคราะห์ข้อมูลและสรุปผลการทำงานจากแบบบันทึกผลปริมาณจำนวนงานของกลุ่มตัวอย่างจากการทำงานด้วยวิธีการแบบแมนวล (Manual) และออโตเมชันตามแนวคิด CI/CD ของกระบวนการ DevOps เพื่อวัดผลความสำเร็จตามข้อตกลงระดับบริการที่กำหนดเพื่อสนับสนุนข้อตกลงระดับการบริการทางธุรกิจ

3.2 ขั้นตอนการเก็บรวบรวมข้อมูลแบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ

3.2.1 นำชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สาธิตการทำงานแก่กลุ่มตัวอย่าง

3.2.2 ใช้เครื่องมือแบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ เพื่อให้กลุ่มตัวอย่างทำการประเมินการใช้งานชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ตามคำถามการประเมินการใช้งานภายในเครื่องมือ

3.2.3 แจกแจงคะแนนและความถี่ของเครื่องมือ โดยนำข้อมูลเข้าสู่กระบวนการวิเคราะห์ข้อมูลด้วยโปรแกรมวิเคราะห์ข้อมูลทางสถิติ และสรุปผล ประกอบด้วย 3 ส่วน คือ

1) ตอนที่ 1 ข้อมูลทั่วไปสำหรับผู้ตอบแบบประเมินการใช้งาน

2) ตอนที่ 2 แบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ โดยแบ่งเป็น 3 ด้าน ดังนี้

2.1) การใช้งานและประสิทธิภาพของออโตเมชันเซิร์ฟเวอร์

2.2) การลดระยะเวลาและขั้นตอนการทำงานจากการใช้งาน

2.3) การทำงานตามเป้าหมายของ SLA

3) ตอนที่ 3 ข้อเสนอแนะ หรือ ความคิดเห็น

4. วิธีการวิเคราะห์และนำเสนอข้อมูล

ในส่วนของการวิเคราะห์ข้อมูลของงานวิจัยนี้จะมีการวิเคราะห์ข้อมูลแบ่งเป็น 2 การวิเคราะห์และนำเสนอข้อมูล โดยใช้โปรแกรมสำเร็จรูปทางสถิติในการวิเคราะห์ข้อมูล ดังนี้

4.1 การวิเคราะห์แบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน Service Level Agreements ทางธุรกิจ โดยกลุ่มตัวอย่าง ซึ่งจะทำการวิเคราะห์ข้อมูลด้วยสูตรคำนวณทางสถิติเพื่อวิเคราะห์ข้อมูล ดังนี้

4.2.1 ค่าร้อยละ (Percentage) ใช้ในการวิเคราะห์ข้อมูลทั่วไปสำหรับผู้ตอบแบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ โดยทำการแจกแจงเป็นค่าความถี่ร้อยละ ด้วยสูตรคำนวณดังสมการที่ 2

$$A = \frac{(X \times 100)}{N} \quad (2)$$

เมื่อ A แทน จำนวนร้อยละของผู้ประเมินการใช้งาน

X แทน จำนวนความถี่ผู้ประเมินการใช้งานแต่ละข้อ

N แทน จำนวนผู้ประเมินการใช้งานทั้งหมด

4.2.2 ค่าเฉลี่ย (Arithmetic mean) หรือ $\bar{X}$ ใช้ในการวิเคราะห์ข้อมูลเพื่อหาค่าเฉลี่ยโดยรวมของการประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ ในแต่ละข้อ และนำค่าที่ได้มาใช้ในการเทียบระดับคะแนนตามที่กำหนดไว้ 5 ระดับ ดังนี้

ค่าเฉลี่ย ($\bar{X}$) 4.51 – 5.00 หมายถึง ระดับมากที่สุด

ค่าเฉลี่ย ($\bar{X}$) 3.51 – 4.50 หมายถึง ระดับมาก

ค่าเฉลี่ย ($\bar{X}$) 2.51 – 3.50 หมายถึง ระดับปานกลาง

ค่าเฉลี่ย ($\bar{X}$) 1.51 – 2.50 หมายถึง ระดับน้อย

ค่าเฉลี่ย ($\bar{X}$) 1.00 – 1.50 หมายถึง ระดับน้อยที่สุด

โดยใช้สูตรการคำนวณทางสถิติเพื่อหาค่าเฉลี่ยในการวิเคราะห์ข้อมูล ดังนี้

$$(\bar{X}) = \frac{\sum fx}{n} \quad (3)$$

เมื่อ $(\bar{X})$ แทน ค่าเฉลี่ยที่ได้จากการคำนวณ

n แทน จำนวนผู้ประเมินการใช้งานทั้งหมด

$\sum fx$ แทน จำนวนผลรวมของค่าความถี่การประเมินแต่ละค่า

4.2.3 ค่าเบี่ยงเบนมาตรฐาน (Standard Division) หรือ S.D. ใช้ในการวิเคราะห์ข้อมูลเพื่อหาค่าความแปรปรวนระดับนัยสำคัญของความน่าจะเป็นของกลุ่มตัวอย่างการประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน Service Level Agreements ทางธุรกิจ ในแต่ละข้อ โดยใช้สูตรการคำนวณทางสถิติเพื่อใช้ในการวิเคราะห์ข้อมูล ดังนี้

$$S.D. = \sqrt{\frac{n\sum x^2 - (\sum x)^2}{n(n-1)}} \quad (4)$$

เมื่อ S. D. แทน ค่าเบี่ยงเบนมาตรฐาน

x แทน ค่าข้อมูลแต่ละลำดับค่า

n แทน จำนวนทั้งหมดของกลุ่มตัวอย่าง

บทที่ 4
ผลการดำเนินการวิจัย

เมื่อได้ดำเนินการวิจัยตามขั้นตอนวิธีการดำเนินการวิจัยเรียบร้อยแล้วนั้น ในส่วนการนำเสนอผลการดำเนินการวิจัยของวิทยานิพนธ์ ซึ่งสอดคล้องกับวัตถุประสงค์งานวิจัยดังต่อไปนี้

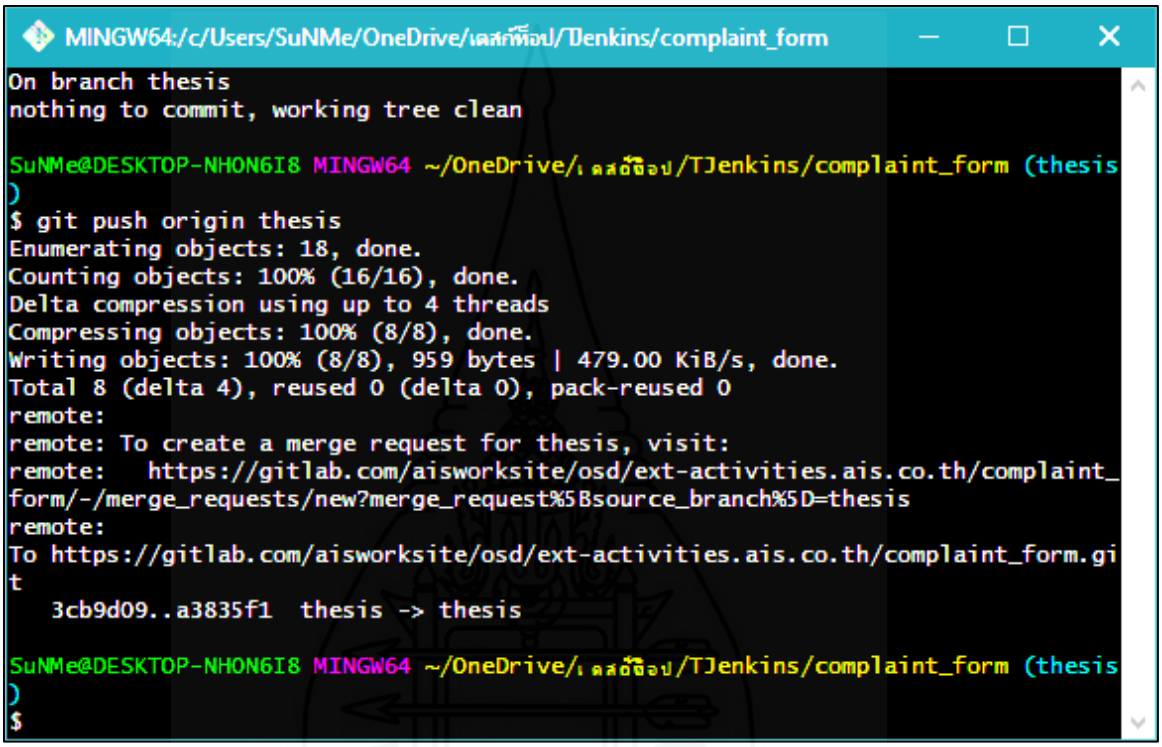
1. การพัฒนาออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ
2. ผลการวิเคราะห์ประสิทธิภาพของออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ
3. ผลการประเมินผลการใช้งานของออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ

1. ผลการพัฒนาออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ

ในหัวข้อนี้จะแสดงให้เห็นถึงวิธีการและขั้นตอนในการพัฒนาออโตเมชันเซิร์ฟเวอร์ตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ ซึ่งเป็นการพัฒนาชุดคำสั่งเพื่อให้ทำงานตามรูปแบบอัตโนมัติภายในบริบทการทำงานของหน่วยงานกลุ่มตัวอย่าง เพื่อให้บรรลุเป้าหมายการทำงานในการสนับสนุนตามข้อตกลงระดับการบริการทางธุรกิจ เพื่อใช้ในการเก็บผลและเปรียบเทียบผลการทำงานตั้งแต่กระบวนการหลังจาก

นักพัฒนาได้ทำการพัฒนาและแก้ไขปัญหาต่างๆ ก่อนดำเนินการตามกระบวนการต่างๆในการส่งมอบและปรับใช้ต่อไป ซึ่งแสดงเป็นกระบวนการต่างๆดังนี้

1.1 แหล่งจัดการชุดคำสั่ง (Source Control) ด้วย GitLab



ภาพที่ 4.1 ตัวอย่างการสั่งการกระบวนการบริหารจัดการชุดคำสั่งด้วย GitLab โดยเครื่องมือ Git Bash

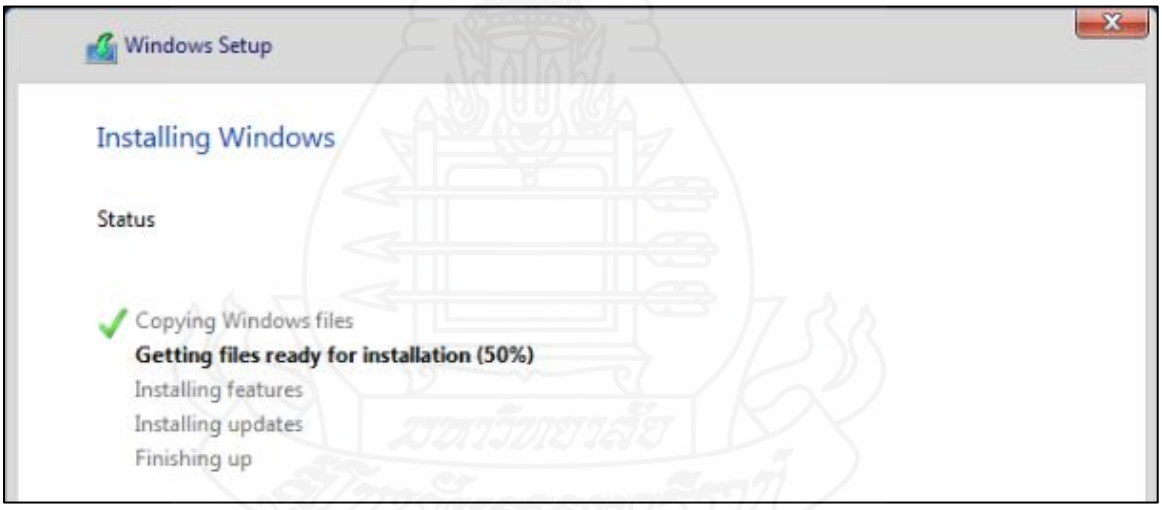
จากภาพที่ 4.1 เป็นคำสั่งการทำการพุช (Push) ชุดคำสั่ง หลังการคอมมิท (Commit) ชุดคำสั่ง เพื่อรีโมทข้อมูลชุดคำสั่งขึ้นสู่กระบวนการบริหารจัดการด้วย GitLab ไปยัง บรานซ์ (Branch) ทำต้องการจัดเก็บ โดยอาศัยเครื่องมือ Git Bash ทำหน้าที่ในการรับคำสั่งการทำงาน และบันทึกผลเวลาที่ได้ทำการพุช (Push) ด้วยคำสั่ง git push origin thesis เพื่อบอกให้ Git Bash ทำหน้าที่นำชุดคำสั่งจัดเก็บไปยัง thesis branch โดย Git Bash จะทำการแสดงผลการทำงานแต่ละขึ้นตั้งแต่ขั้นตอนการบีบอัดและส่งชุดคำสั่งขึ้นบน GitLab ตาม Path และ URL ที่ได้ทำการสร้างการติดตามการเปลี่ยนแปลงชุดคำสั่งไว้ เมื่อเสร็จสิ้นกระบวนการจะแสดงลำดับการคอมมิทด้วยชุดตัวเลขลำดับและรายงานการส่งชุดคำสั่ง เพื่อแจ้งให้ผู้ส่งคำสั่งทราบว่าส่งชุดคำสั่งจากเครื่องที่พัฒนาในบรานซ์ใดไป

จัดเก็บบนเครื่องรีโมทเซิร์ฟเวอร์ที่บรานซ์ใด ซึ่งจากภาพแสดงเป็น thesis -> thesis คือส่งชุดคำสั่งจากเครื่องพัฒนาในบรานซ์ thesis ไปยังเครื่องรีโมทเซิร์ฟเวอร์ที่บรานซ์ thesis

1.2 การทำงานตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ

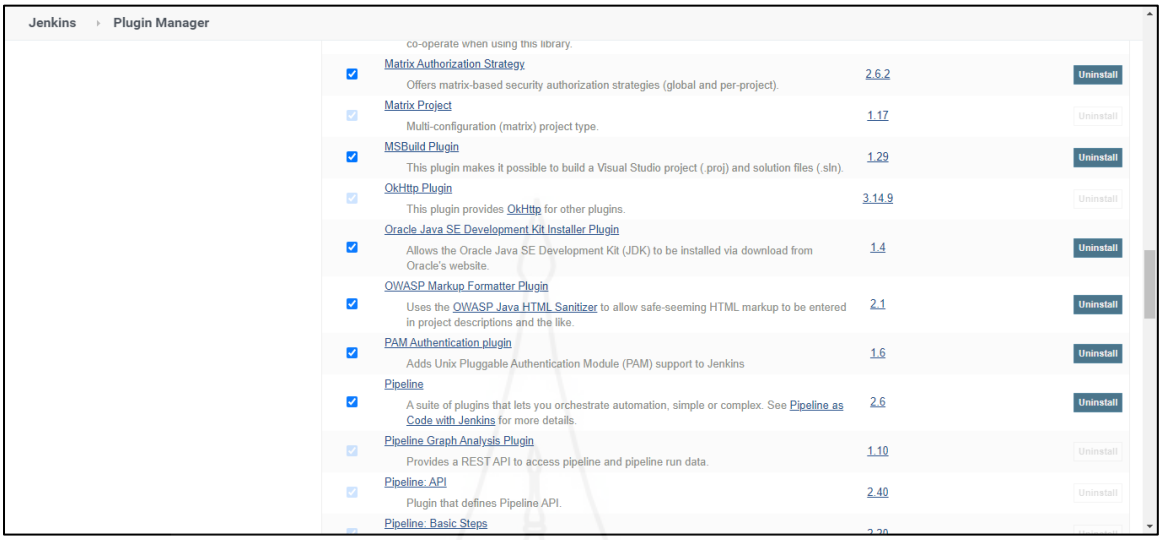
การพัฒนาออโตเมชันเซิร์ฟเวอร์ตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ โดยอาศัยเครื่องมือต่างๆ ได้แก่

1.2.1 Microsoft Windows Server 2019 Datacenter ซึ่งในกระบวนการทำการวิจัยนี้ ได้ทำการใช้งานด้วย Student Version เพื่อทำการวิจัยและเรียนรู้การทำงานของระบบปฏิบัติการ Microsoft Windows Server 2019 Datacenter ที่สามารถนำมาทดลองใช้งานได้สูงสุด 180 วัน ต่อการติดตั้ง 1 ครั้ง โดยจากภาพที่ 4.2 แสดงให้เห็นถึงการทำงานของขั้นตอนแต่ละขั้นตอนของการติดตั้ง



ภาพที่ 4.2 กระบวนการติดตั้ง Windows Server 2019 Datacenter Student Version

1.2.2 Jenkins Open source เพื่อเป็นเครื่องมือช่วยในการพัฒนาการทำงานในรูปแบบของออโตเมชันเซิร์ฟเวอร์ ซึ่งต้องทำการติดตั้ง Plugin เพิ่มเติมสำหรับการทำงานร่วมกับ Microsoft Windows Server 2019 Datacenter ได้แก่ MS-Build Plugin และ Pipeline โดย Plugin อื่นๆที่จำเป็น Jenkins จะทำการดาวน์โหลดและติดตั้งพร้อมกับขั้นตอนการติดตั้ง Jenkins แล้ว



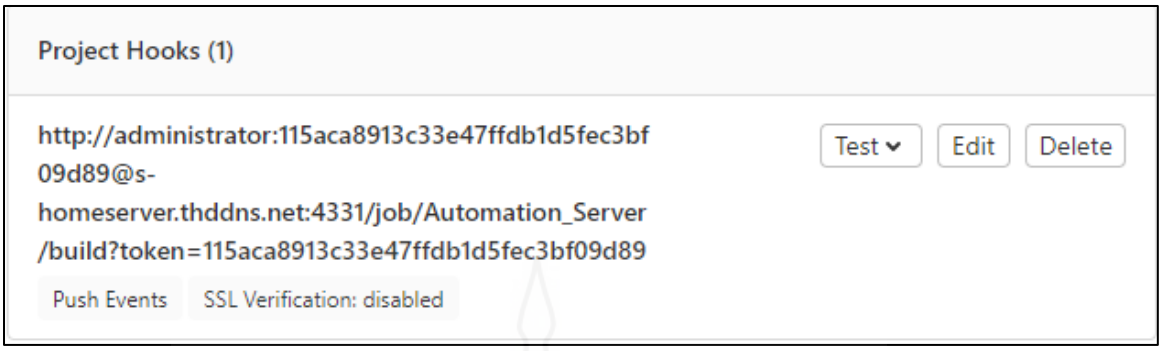
ภาพที่ 4.3 Jenkins Plugin Manager

1.2.3 Pipeline Script สำหรับการสั่งการเพื่อให้ Jenkins ทำงานร่วมกับชุดคำสั่ง Pipeline ตามขั้นตอนที่กำหนดในรูปแบบของการทำงานตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ ซึ่งเป็นการเขียนด้วยภาษา Groovy ร่วมกับคำสั่ง Unix และ/หรือ Command Line เนื่องจากในบางขั้นตอนต้องอาศัย Command Line ในการสั่งการให้ Windows Server สามารถทำงานตามรูปแบบของบริบทการทำงานของหน่วยงาน ซึ่งประกอบไปด้วยขั้นตอนของการสร้างและจัดกลุ่มโปรแกรมสำเร็จ (Build and Package), เผยแพร่ (Release) และติดตั้ง (Deploy) ตามลำดับ ทั้งนี้ขั้นตอนต่างๆสามารถปรับเปลี่ยนได้ตามบริบทของแต่ละหน่วยงาน ซึ่งจะต้องเขียนชุดคำสั่งเพื่อบังคับการทำงานด้วยตนเองเช่นกัน โดยการเขียน 1 ไฟล์ Pipeline Script จะสามารถใช้งานได้กับโครงการเดียวกันเท่านั้น เนื่องจากต้องระบุชื่อของ Solution File และตำแหน่งในการปรับใช้ที่ต่างกัน หากมีโครงการที่ต่างกันไปจะต้องสร้าง Pipeline Script ขึ้นใหม่ และปรับเปลี่ยนไปตามชื่อของไฟล์ตั้งต้นของการทำงาน (Solution File) และตำแหน่งที่ต้องการปรับใช้งานจริง ทั้งนี้ในการวิจัยครั้งนี้เป็นการทำงานอยู่บนพื้นฐานการทำงานของ .Net Framework ที่ต้องอาศัย MS-Build ช่วยในการสร้างโครงการ

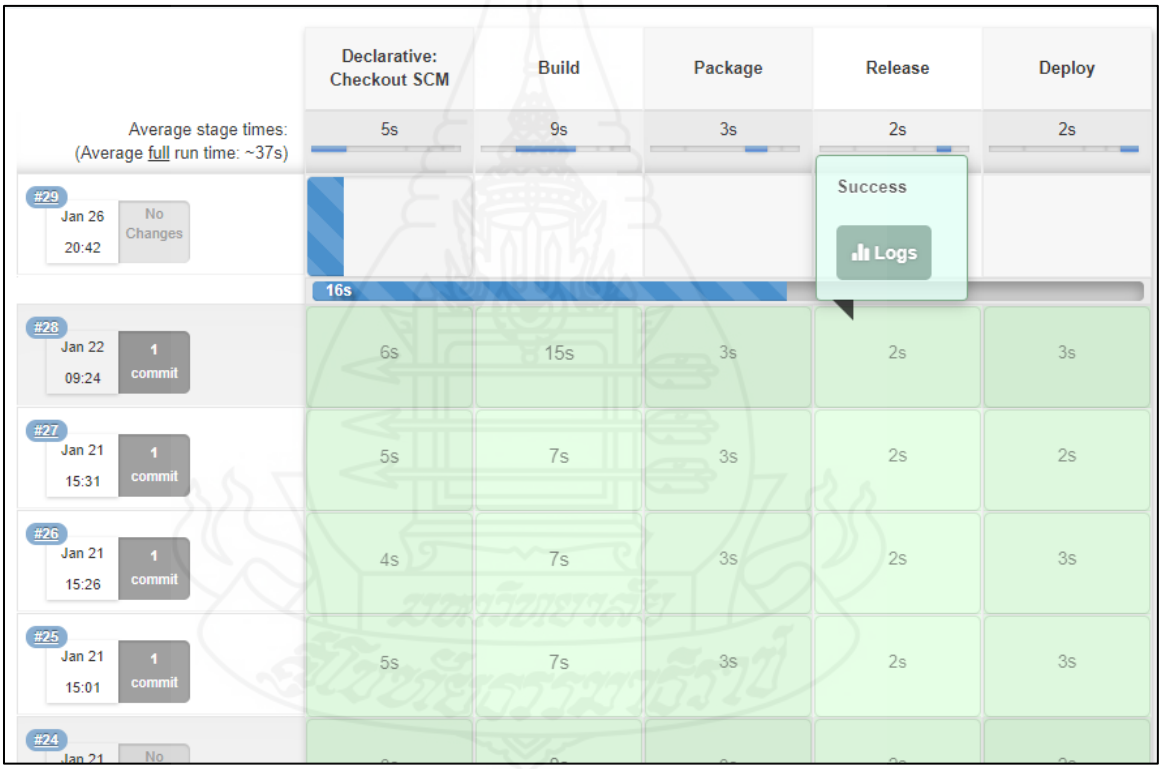
```
pipeline{
    agent any // เรียก agent ที่จะทำงาน job jenkins นี้ ในที่นี้กำหนดเป็น any คือ เครื่องไหนก็ได้ที่อยู่ภายใต้ jenkins master เครื่องนี้

    environment{ // ในส่วน environment เป็นการประกาศตัวแปร
        //LINE_TOKEN = credentials('line-notify')

        softwareHome = 'c:\\inetpub\\wwwroot'
        buildVersion = ''
        buildSolutionConfigure = 'Debug'
        buildProject = 'Complaint_Form'
        buildSolutionFile = 'Complaint_Form.sln'
        buildMvcVersion = ''
        processDir = 'D:\\'

}// ในส่วน environment เป็นการประกาศตัวแปร

    stages{ // เป็นการเปิด stage ในการทำงาน
        stage('Build'){ // stage ชื่อว่า Build
            steps {
                bat "\"${tool 'DotNetx86'}\" ${buildSolutionFile} /p:VisualStudioVersion=15.0 /p:C

                    }
        }
        stage('Package'){ // stage ชื่อว่า Package
            steps{
```

ภาพที่ 4.4 ตัวอย่างคำสั่ง Pipeline Script

1.2.4 สร้างการทำงานร่วมกันระหว่าง Jenkins และแหล่งจัดเก็บชุดคำสั่ง (Source Control) ด้วย GitLab ซึ่งกระบวนการนี้จะทำให้ออโตเมชันเซิร์ฟเวอร์สามารถทำงานได้อย่างอัตโนมัติหลังจากที่นักพัฒนาได้ทำการดำเนินการตามกระบวนการบริหารจัดการชุดคำสั่ง (Source Control) ด้วย GitLab เรียบร้อยแล้ว ซึ่งจะเป็นการสั่งการทำงานให้ Jenkins เริ่มทำการสร้างงาน (Build Job) ตามขั้นตอนที่ได้เขียน Pipeline Script ไว้ผ่าน Jenkins Webhook ของแต่ละโครงการ เพื่อเข้าสู่กระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ ให้บรรลุวัตถุประสงค์ในการสนับสนุนข้อตกลงระดับการบริการทางธุรกิจที่กำหนดไว้ ซึ่งหากสามารถเรียกด้วย Jenkins Webhook ได้สำเร็จ Jenkins จะการสร้างและจัดกลุ่มโปรแกรมสำเร็จ (Build and Package), เผยแพร่ (Release) และติดตั้ง (Deploy) ไปยังตำแหน่งการปรับใช้งานจริงที่ได้กำหนดไว้ใน Pipeline Script ซึ่งจะแสดงระยะเวลาในการประมวลผลในแต่ละขั้นตอนการทำงาน ซึ่งจะนำไปใช้ในการประเมินประสิทธิภาพการทำงานของชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ต่อไป ดังภาพที่ 4.5 และ 4.6



ภาพที่ 4.5 ตัวอย่างการทำงานร่วมกันระหว่าง Jenkins และ Source Control ผ่าน Webhook



ภาพที่ 4.6 ตัวอย่างการทำงานตามขั้นตอนใน Pipeline Script ด้วย Jenkins

2. ผลการวิเคราะห์ประสิทธิภาพของออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ

ในหัวข้อนี้จะแสดงให้เห็นถึงวิธีการเปรียบเทียบประสิทธิภาพในการใช้งานระหว่างการทำงานด้วยกระบวนการแบบแมนนวลด้วยระยะเวลาที่เกิดขึ้นบนซอฟต์แวร์เว็บแอปพลิเคชัน สำหรับใช้ติดตามการทำงานตามกระบวนการติดตั้งระบบ (Deployment) ของหน่วยงานและกระบวนการทำงานแบบอัตโนมัติตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ ด้วยชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps โดยมีสมมติฐานและวัตถุประสงค์เพื่อให้บรรลุเป้าหมายการทำงานในการสนับสนุนตามข้อตกลงระดับการบริการทางธุรกิจ ซึ่งจะเริ่มทำการเก็บผลและเปรียบเทียบผลการทำงานตั้งแต่กระบวนการหลังจากนักพัฒนาได้ทำการพัฒนาและแก้ไขปัญหาต่างๆ ก่อนดำเนินการตามกระบวนการต่างๆในการส่งมอบและปรับใช้ต่อไป โดยจากการศึกษาวิจัยและเก็บผลการวิจัยได้ดังนี้

2.1 ผลสรุปความสำเร็จเพื่อสนับสนุนข้อตกลงระดับการบริการทางธุรกิจตามปริมาณงานแยกตามขนาดของประเภทซอฟต์แวร์หรือแอปพลิเคชันด้วยการแมนนวล

เป็นการแสดงถึงผลสรุปของระดับความสำเร็จเพื่อสนับสนุนข้อตกลงระดับการบริการทางธุรกิจ (SLA) โดยแยกตามปริมาณงานด้วยขนาดของซอฟต์แวร์หรือแอปพลิเคชันที่ถูกกำหนดขึ้นจากเอกสารการขอปรับปรุงแก้ไขระบบ ซึ่งหน่วยงานได้มีการกำหนดขึ้นไว้แล้ว โดยอาศัยหลักทฤษฎีของ COCOMO ในการกำหนดขนาดของซอฟต์แวร์ และแสดงผลสรุปขึ้นจากการดำเนินการวิจัยตามขั้นตอนการวิจัย ภายใต้หัวข้อ การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps ซึ่งในส่วนนี้จะแสดงให้เห็นถึงความสำเร็จที่เกิดขึ้นจากกระบวนการการทำงานและประเมินประสิทธิภาพของการทำงานแบบขั้นตอนแบบแมนนวล โดยอาศัยการทำงานด้วยมือและเครื่องมือซอฟต์แวร์ติดตามผลทางธุรกิจ สามารถสรุปข้อมูลได้ดังนี้ ขนาดซอฟต์แวร์ M ปริมาณ 1 รายการคิดเป็น ร้อยละ 1.16 ของจำนวนรายการ 86 รายการจากการดำเนินการแบบแมนนวล สำเร็จผลตามระยะข้อตกลงระดับบริการที่กำหนด และปริมาณ 85 รายการจาก 86 รายการ คิดเป็นร้อยละ 98.84 ไม่สำเร็จผลตามระยะข้อตกลงระดับบริการที่กำหนด ซึ่งทำให้ปรากฏผลความสำเร็จขึ้นตามตารางที่ 4.1

ตารางที่ 4.1 ปริมาณงานตามขนาดของประเภทซอฟต์แวร์หรือแอปพลิเคชันด้วยการแมนนวล

ขนาด	สำเร็จตาม SLA		ไม่สำเร็จตาม SLA		รวม	
ซอฟต์แวร์	ปริมาณ	ร้อยละ	ปริมาณ	ร้อยละ	ปริมาณ	ร้อยละ
S	0	0	5	5.81	5	5.81
M	1	1.16	44	51.17	45	52.33
L	0	0	36	41.86	36	41.86
รวม	1	1.16	85	98.84	86	100

2.2 ผลสรุปความสำเร็จเพื่อสนับสนุนข้อตกลงระดับการบริการทางธุรกิจตามปริมาณงานแยกตามขนาดของประเภทซอฟต์แวร์หรือแอปพลิเคชันด้วยออโตเมชันเซิร์ฟเวอร์

เป็นการแสดงถึงผลสรุปของระดับความสำเร็จเพื่อสนับสนุนข้อตกลงระดับการบริการทางธุรกิจ (SLA) โดยแยกตามปริมาณงานด้วยขนาดของซอฟต์แวร์หรือแอปพลิเคชัน ซึ่งในส่วนนี้จะแสดงให้เห็นถึงความสำเร็จที่เกิดขึ้นจากกระบวนการการทำงานและประเมินประสิทธิภาพของการทำงานแบบใช้แนวคิด CI/CD ภายใต้กระบวนการ DevOps โดยอาศัยชุดการทำงานออโตเมชันเซิร์ฟเวอร์สามารถสรุปข้อมูลได้ดังนี้ อัตราความสำเร็จตามข้อตกลงระดับบริการแยกตามขนาดซอฟต์แวร์จากรายการทำงานทั้งสิ้น 86 รายการ สำเร็จตามเป้าหมายระยะข้อตกลงระดับบริการคิดเป็นร้อยละ 100 ตามขนาดของซอฟต์แวร์ M, L,S ในอัตราส่วนร้อยละ 52.33, 41.86 และ 5.81 ตามลำดับ ในขณะนี้อัตราส่วนความไม่สำเร็จตามข้อตกลงระดับบริการอยู่ที่ร้อยละ 0 หรือหมายความถึงการวิจัยตามวิธีการวิจัยในครั้งนี้จากประชากรและกลุ่มตัวอย่าง ไม่พบว่าการทำงานด้วยชุดเครื่องมือออโตเมชันเซิร์ฟเวอร์ทำให้การทำงานไม่สำเร็จตามระยะข้อตกลงระดับบริการที่กำหนดไว้ แต่เนื่องจากในการทดสอบได้ทำการทดสอบด้วยชุดคำสั่งที่ผ่านการทดสอบแบบหน่วยทดสอบ (Unit Testing) โดยนักพัฒนาแล้ว รวมถึงโครงสร้างสถานะเครือข่ายและสภาพแวดล้อมที่ใช้ทดสอบสามารถทำงานได้อย่างสมบูรณ์ในทุกการทดสอบ แต่หากการทดสอบดังกล่าวเกิดปัญหาที่เกี่ยวข้องกับกระบวนการสร้างและจัดกลุ่มโปรแกรมสำเร็จ (Build and Package) ด้วย MS-Build หรือมีสภาพแวดล้อมทางโครงสร้างรวมถึงสถานะเครือข่ายที่ติดขัด ก็อาจทำให้ผลการทดสอบการประสิทธิภาพและการทำงานของออโตเมชันเซิร์ฟเวอร์นั้น ไม่สมบูรณ์ตามผลการทดสอบประสิทธิภาพที่ปรากฏขึ้นตามตารางที่ 4.2 ได้

ตารางที่ 4.2 ปริมาณงานตามขนาดของประเภทซอฟต์แวร์หรือแอปพลิเคชันด้วยออโตเมชันเซิร์ฟเวอร์

ขนาด	สำเร็จตาม SLA		ไม่สำเร็จตาม SLA		รวม	
ซอฟต์แวร์	ปริมาณ	ร้อยละ	ปริมาณ	ร้อยละ	ปริมาณ	ร้อยละ
S	5	5.81	0	0	5	5.81
M	45	52.33	0	0	45	52.33
L	36	41.86	0	0	36	41.86
รวม	86	100	0	0	86	100

2.3 ผลสรุปเปรียบเทียบการประเมินประสิทธิภาพการใช้งานของออโตเมชันเซิร์ฟเวอร์ด้วยชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps

เป็นการแสดงผลเพื่อแสดงให้เห็นถึงประสิทธิภาพการทำงานตามวิธีการดำเนินงานวิจัย เรื่อง การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps ที่เปรียบเทียบกันระหว่างระดับเปอร์เซ็นต์ความสำเร็จตามข้อตกลงระดับบริการและเครื่องมือที่ใช้ในการทำงานเพื่อประเมินประสิทธิภาพตามขั้นตอนการดำเนินการวิจัยที่ได้กำหนดไว้ ซึ่งแสดงให้เห็นว่าการทำงานด้วยชุดทำงานออโตเมชันเซิร์ฟเวอร์ช่วยให้ระยะเวลาการสนับสนุนข้อตกลงระดับการบริการทางธุรกิจสำเร็จตามระยะเวลาที่กำหนดไว้ ร้อยละ 100 ในขณะที่การทำงานแบบขั้นตอนแมนนวลทำให้การสนับสนุนข้อตกลงระดับการบริการทางธุรกิจสำเร็จตามระยะข้อตกลงระดับบริการที่กำหนดเพียง ร้อยละ 1.16 ไม่ได้สำเร็จตามระยะข้อตกลงระดับบริการถึงร้อยละ 98.84 ตามตารางที่ 4.3

ตารางที่ 4.3 ผลสรุปการประเมินประสิทธิภาพการใช้งานของออโตเมชันเซิร์ฟเวอร์

Tools	สำเร็จตาม SLA		ไม่สำเร็จตาม SLA		รวม	
	ปริมาณ	ร้อยละ	ปริมาณ	ร้อยละ	ปริมาณ	ร้อยละ
Automation Server	86	100	0	0	86	100
Manual Step	1	1.16	85	98.84	86	100

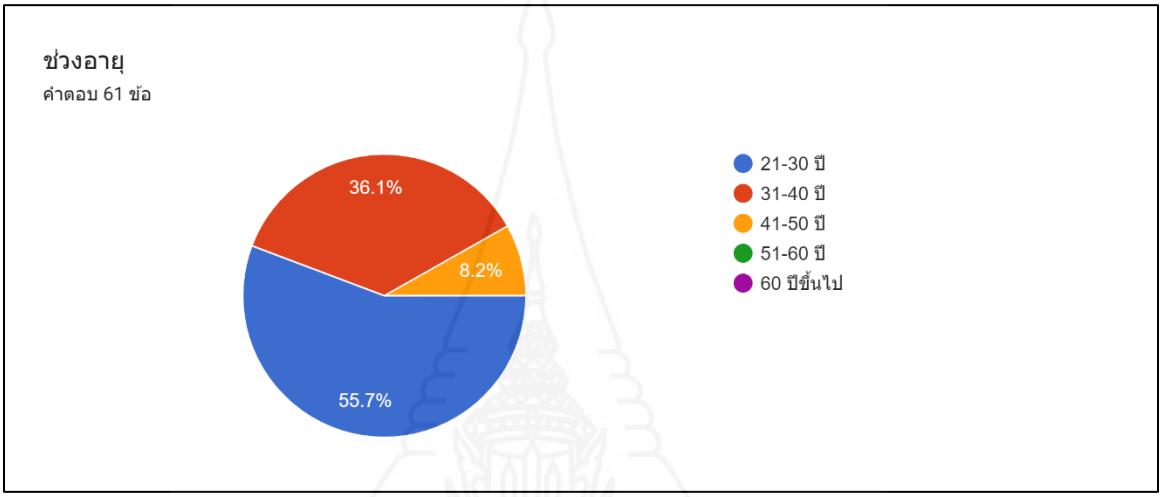
3. ผลการประเมินผลการใช้งานของออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ

ในหัวข้อนี้จะแสดงให้เห็นถึงผลการประเมินการใช้งานของผู้ใช้งานตามประชากรและกลุ่มตัวอย่างที่กำหนดขึ้นให้ทำการทดสอบการประเมินประสิทธิภาพการใช้งานของออโตเมชันเซิร์ฟเวอร์ด้วยชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps โดยใช้ Google Form เป็นเครื่องมือใช้ในการเก็บรวบรวมข้อมูล แปลผล และสรุปผล จากประชากรทั้งสิ้น 86 รายการ แต่มีจำนวนกลุ่มตัวอย่างที่ตอบแบบประเมินทั้งสิ้น 61 คน เนื่องจากมีการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน Service Level Agreements ทางธุรกิจบางรายการ ดำเนินการทดสอบและประเมินประสิทธิภาพการใช้งานมากกว่า 1 ครั้งหรือมากกว่า 1 ซอฟต์แวร์ ใน 1 คน จึงทำให้ผลการตอบแบบประเมินการใช้งานได้ไม่เท่ากับจำนวนครั้งในทดสอบประสิทธิภาพ ซึ่งแบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน Service Level Agreements ทางธุรกิจ ได้แบ่งออกเป็น 3 ตอน ดังนี้

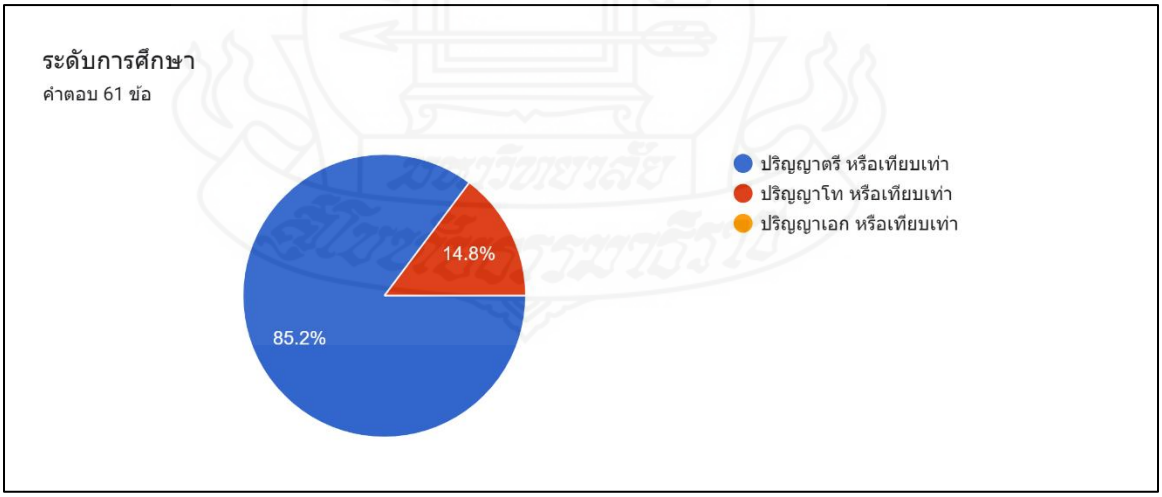
3.1 ตอนที่ 1 ข้อมูลทั่วไปสำหรับผู้ตอบแบบประเมินการใช้งาน

เป็นการแสดงให้เห็นถึงปริมาณและร้อยละของผู้ตอบแบบประเมินการใช้งานโดยแยกเป็นกลุ่มคำถาม ช่วงอายุ ระดับการศึกษา และตำแหน่ง/สายงาน ตามลำดับ โดยอัตราส่วนของผู้ตอบแบบสอบถาม 3 อันดับแรกได้แก่ ช่วงอายุ 21-30 ปี, 31-40 ปี และ 41-50 ปี ด้วยอัตราส่วนร้อยละ 55.70, 36.10 และ 8.20 ตามลำดับช่วงอายุที่ได้กล่าวไว้ และปรากฏตามภาพที่ 4.7 ในส่วนของอัตราส่วนระดับการศึกษา 3 อันดับแรก ได้แก่ ระดับปริญญาตรีหรือเทียบเท่า, ระดับปริญญาโทหรือเทียบเท่า และระดับปริญญาเอกหรือเท่าเทียบ โดยคิดเป็นอัตราส่วนร้อยละ 85.20, 14.80 และ 0 ตามลำดับ ซึ่งปรากฏตามภาพที่ 4.8 และในอัตราส่วนของผู้ตอบแบบประเมินการใช้งานโดยแยกตามตำแหน่ง/สายงาน 3 อันดับแรก ได้แก่ Programmer / Developer / Software Engineer ซึ่งทำหน้าที่ในการพัฒนาระบบงานที่ต้องทำหน้าที่แก้ไขหรือเขียนโปรแกรมเพื่อแก้ปัญหาระบบให้เสร็จสิ้นทันตามกำหนดระยะเวลาของข้อตกลงระดับบริการที่กำหนดไว้, Tester / Automation Tester / Software QA ซึ่งทำหน้าที่ในการตรวจสอบคุณภาพ ทดสอบระบบ และแจ้งปัญหาที่เกิดขึ้นกับระบบให้แก้นักพัฒนา

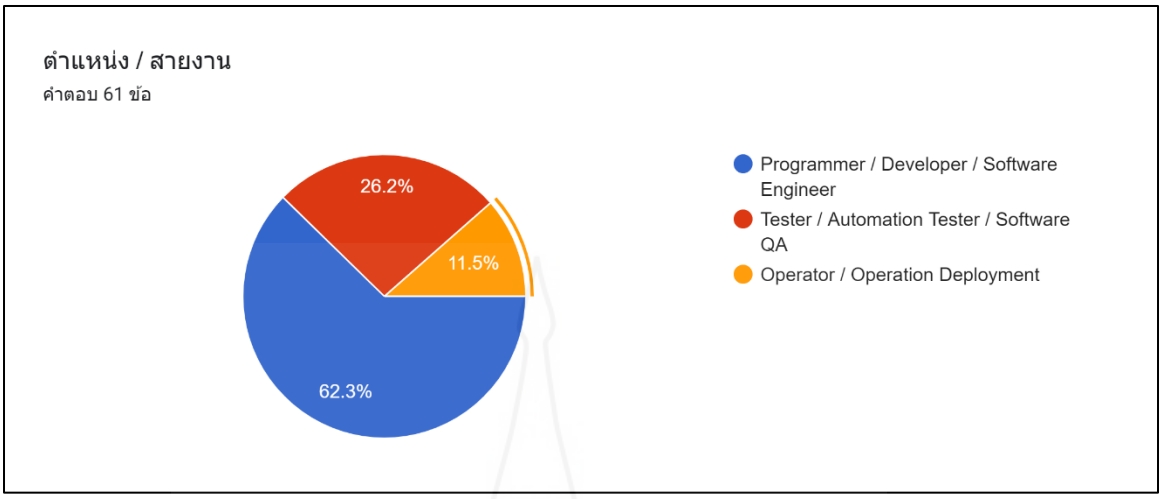
เพื่อทำการแก้ไขปัญหา อันมีส่วนกับการทำงานที่ส่งผลต่อข้อตกลงระดับบริการทางธุรกิจ และ Operator / Operation Deployment ซึ่งทำหน้าที่ในการเป็นผู้ทำหน้าที่ในการติดตั้งระบบบนเครื่องแม่ข่าย รวมถึงติดตามและตรวจสอบสภาพการทำงานของเครื่องแม่ข่ายที่ซอฟต์แวร์ระบบที่นำไปติดตั้งได้ใช้งาน ในอัตราส่วนร้อยละ 62.30, 26.20 และ 11.50 ตามลำดับ ซึ่งปรากฏไว้ตามภาพที่ 4.9



ภาพที่ 4.7 อัตราส่วนผู้ตอบแบบประเมินการใช้งานตามช่วงอายุ



ภาพที่ 4.8 อัตราส่วนผู้ตอบแบบประเมินการใช้งานตามระดับการศึกษา



ภาพที่ 4.9 อัตราส่วนผู้ตอบแบบประเมินการใช้งานตามตำแหน่ง/สายงาน

3.2 ตอนที่ 2 แบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ

เป็นการแสดงถึงผลการประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ ที่ผู้ทดสอบประสิทธิภาพได้ทำการทดสอบประสิทธิภาพและประเมินผลการใช้งาน โดย มีระดับการให้คะแนนตอบคำถาม 5 ระดับ ดังนี้

คะแนน 5 หมายถึง ระดับมากที่สุด
คะแนน 4 หมายถึง ระดับมาก
คะแนน 3 หมายถึง ระดับปานกลาง
คะแนน 2 หมายถึง ระดับน้อย
คะแนน 1 หมายถึง ระดับน้อยที่สุด

และเกณฑ์ในการประเมินผลการใช้งานได้แบ่งเป็น 5 ระดับ ดังนี้

ค่าเฉลี่ย ($\bar{X}$) 4.51 – 5.00 หมายถึง ระดับมากที่สุด
ค่าเฉลี่ย ($\bar{X}$) 3.51 – 4.50 หมายถึง ระดับมาก
ค่าเฉลี่ย ($\bar{X}$) 2.51 – 3.50 หมายถึง ระดับปานกลาง
ค่าเฉลี่ย ($\bar{X}$) 1.51 – 2.50 หมายถึง ระดับน้อย

ค่าเฉลี่ย ($\bar{X}$) 1.00 – 1.50 หมายถึง ระดับน้อยที่สุด

และแบ่งหมวดคำถามที่ใช้สอบถามเพื่อประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ เป็น 3 ด้าน และแสดงอัตราส่วนร้อยละในการให้ระดับคะแนนของผู้ตอบแบบประเมินการใช้งาน ดังนี้

ตารางที่ 4.4 แบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจ ตอนที่ 2

ลำดับ	รายการสอบถาม	ผลประเมินการใช้งาน		
		$\bar{X}$	*SD*	ระดับการประเมิน
1	**การใช้งานและประสิทธิภาพของออโตเมชันเซิร์ฟเวอร์**			
	1.1 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถทำงานได้ตามขั้นตอนที่กำหนดไว้	4.0820	0.66571	มาก
	1.2 Automation Server ภายใต้แนวคิด CI/CD สามารถนำไปใช้งานในการทำงานได้จริงภายใต้กระบวนการ DevOps	4.1639	0.58253	มาก
	1.3 Automation Server ภายใต้แนวคิด CI/CD สามารถใช้งานทดแทนรูปแบบการทำงาน Manual ได้	4.0984	0.76822	มาก
	1.4 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถเพิ่มประสิทธิภาพในการทำงานได้จากรูปแบบการทำงาน Manual	4.0328	0.81583	มาก
	1.5 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถนำไปต่อยอดเพื่อประยุกต์ใช้ร่วมกับกลยุทธ์การบริหารจัดการทางเทคโนโลยีสารสนเทศได้	4.1475	0.72655	มาก

ลำดับ	รายการสอบถาม	ผลประเมินการใช้งาน		
		$\bar{X}$	*SD*	ระดับการประเมิน
	1.6 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถทำงานได้อย่างมีประสิทธิภาพ ในขณะที่ใช้จำนวนพนักงานปฏิบัติการน้อยลง	3.9836	0.80606	มาก
	ค่าระดับคะแนนเฉลี่ย ($\bar{X}$) ด้านที่ 1	**4.0847**		**มาก**
	ค่าเบี่ยงเบนมาตรฐาน (SD) ด้านที่ 1	**0.72748**		
2	**การลดระยะเวลาและขั้นตอนการทำงานจากการใช้งาน**			
	2.1 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถช่วยลดระยะเวลาในขั้นตอนการทำงานเพื่อนำไปใช้งานจริง	4.0656	0.67992	มาก
	2.2 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถช่วยลดขั้นตอนการทำงานเพื่อนำไปใช้งานจริง	4.0656	0.74986	มาก
	2.3 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถช่วยลดระยะเวลาในทุกขั้นตอนที่นำมาใช้ทดแทนการทำงานในรูปแบบ Manual	4.0820	0.75928	มาก
	2.4 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ช่วยเพิ่มความสามารถของผลผลิตซอฟต์แวร์ในเชิงกลยุทธ์จากการใช้ระยะเวลาที่ลดลง	4.0000	0.75277	มาก
	2.5 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ช่วยเพิ่มความรวดเร็วให้แก่ผลผลิตซอฟต์แวร์ในการนำออกไปใช้งานกับผู้ใช้งาน	4.0492	0.69345	มาก

ลำดับ	รายการสอบถาม	ผลประเมินการใช้งาน		
		$\bar{X}$	*SD*	ระดับการประเมิน
	ค่าระดับคะแนนเฉลี่ย ($\bar{X}$) ด้านที่ 2	**4.0524**		**มาก**
	ค่าเบี่ยงเบนมาตรฐาน (SD) ด้านที่ 2	**0.72705**		
3	**การทำงานตามเป้าหมายของ Service Level Agreements**			
	3.1 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถทำงานได้ตามเป้าหมายที่กำหนดไว้ใน Service Level Agreements	4.0164	0.67062	มาก
	3.2 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ช่วยเพิ่มเปอร์เซ็นต์ความสำเร็จตาม Service Level Agreements ที่กำหนดไว้	3.9508	0.64359	มาก
	3.3 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ช่วยเพิ่มประสิทธิภาพความสำเร็จจากการทำงานตามแผนงานของฝ่ายงานซึ่งมีส่วนจากเปอร์เซ็นต์ความสำเร็จตาม Service Level Agreements ที่กำหนดไว้	3.8689	0.78476	มาก
	3.4 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ช่วยลดปัญหาที่เกิดขึ้นจากความไม่เป็นไปตามเป้าหมายของ Service Level Agreements ที่กำหนดไว้	3.9016	0.70012	มาก
	3.5 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ทำให้ช่วยทำให้ภาพรวมของการปฏิบัติงานตาม Service Level Agreements ดีขึ้นอย่างเห็นได้ชัด	3.9672	0.65745	มาก
	ค่าระดับคะแนนเฉลี่ย ($\bar{X}$) ด้านที่ 3	**3.9409**		**มาก**

ลำดับ	รายการสอบถาม	ผลประเมินการใช้งาน		
		$\bar{X}$	*SD*	ระดับการประเมิน
	ค่าเบี่ยงเบนมาตรฐาน (SD) ด้านที่ 3	**0.69130**		
	ค่าระดับคะแนนเฉลี่ย ($\bar{X}$) ทุกด้าน	**4.0260**		**มาก**
	ค่าเบี่ยงเบนมาตรฐาน (SD) ทุกด้าน	**0.71527**		

ซึ่งจากระดับคะแนนของผู้ตอบแบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจในทุกหมวดคำถามมีปริมาณอัตราส่วนการให้ระดับคะแนนอยู่ที่ระดับ มาก, มากที่สุด และ ปานกลาง ตามลำดับ โดยมีค่าเฉลี่ยระดับคะแนนในด้านการใช้งานและประสิทธิภาพของออโตเมชันเซิร์ฟเวอร์ในระดับมาก ($\bar{X}$= 4.0847 *SD*= 0.72748) ด้านการลดระยะเวลาและขั้นตอนการทำงานจากการใช้งานในระดับมาก ($\bar{X}$= 4.0524 *SD*= 0.72705) และด้านการทำงานตามเป้าหมายของ Service Level Agreements ในระดับมาก ($\bar{X}$= 3.9409 *SD*= 0.69130) ค่าเฉลี่ยในทุกด้านการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ทางธุรกิจอยู่ในระดับมาก ($\bar{X}$= 4.0260 *SD*= 0.71527)

3.3 ตอนที่ 3 ข้อเสนอแนะ หรือ ความคิดเห็น

เป็นส่วนที่แสดงให้เห็นถึงความคิดเห็นของผู้ตอบแบบประเมินการใช้งาน โดยเป็นคำถามปลายเปิดจำนวน 1 ข้อ เพื่อให้ผู้ตอบแบบสอบถามได้แสดงความคิดเห็นหรือข้อเสนอแนะจากการทดสอบประสิทธิภาพการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA โดยผู้วิจัยได้ทำการสรุปข้อคิดเห็นและข้อเสนอแนะได้ดังนี้

ตารางที่ 4.5 ข้อคิดเห็นหรือหรือเสนอแนะ

ข้อคิดเห็นหรือหรือเสนอแนะ	ปริมาณการตอบ	ร้อยละ
ทำงานได้ดี	3	50.00
ปรับปรุงเพื่อให้ประสิทธิภาพดีขึ้น	1	16.67
ปรับปรุงเพื่อให้สอดคล้องกับบริบทในทุกหน่วยงาน	1	16.67
ทำให้ต้องศึกษาและเรียนรู้เพิ่มเติม*	1	16.66
รวม	**6**	**100**

* หมายถึง ข้อคิดเห็นหรือข้อเสนอแนะไม่สอดคล้องกับการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA หากมีระดับคะแนนที่ต้องปัดระดับคแนน ถือเป็นการปัดลดระดับคะแนนลง

จากตารางที่ 4.5 แสดงให้เห็นว่าผู้ตอบแบบประเมินการใช้งานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน SLA ได้ให้ข้อเสนอแนะและข้อคิดเห็นไว้ โดยสรุปออกมาได้ 4 เรื่องสำคัญ ดังนี้ ทำงานได้ดี, ปรับปรุงเพื่อให้ประสิทธิภาพดีขึ้น, ปรับปรุงเพื่อให้สอดคล้องกับบริบทในทุกหน่วยงาน และทำให้ต้องศึกษาและเรียนรู้เพิ่มเติม ตามอัตราส่วนร้อยละ 50, 16.67, 16.67 และ 16.66 ตามลำดับ

3.4 ผลสรุปสถิติการติดตั้งระบบเพื่อใช้งานจริงของหน่วยงานในการใช้งานและทดสอบประสิทธิภาพด้วยออโตเมชันเซิร์ฟเวอร์

เพื่อให้สอดคล้องกับตารางที่ 1.2 แสดงให้เห็นว่าซอฟต์แวร์แต่ละขนาดที่ใช้ในการทดสอบนั้นมีระยะเวลาในการปรับใช้งาน จากผลการทดสอบประสิทธิภาพอยู่ในระดับนาทีในทุกขนาดซอฟต์แวร์ เนื่องจากในแต่ละพื้นที่การทำงานบนออโตเมชันเซิร์ฟเวอร์นั้น มีระยะเวลาที่ใช้ในการทำงานสูงสุดอยู่ที่ระดับนาที และมีระดับต่ำสุดเป็นระดับวินาที ซึ่งอ้างอิงจากตารางสรุปผลการทดสอบประสิทธิภาพในภาคผนวก ก ทำให้ระยะเวลารวมสูงสุดอยู่ในระดับนาที ในส่วนของความเชื่อมั่นและการตอบสนองของผู้ใช้งานของซอฟต์แวร์ทุกขนาดประเมินจากผลการตอบแบบสอบถามของผู้ใช้งาน โดยความเชื่อมั่นอยู่ที่ระดับสูง เนื่องจากซอฟต์แวร์ที่นำมาใช้ในการทดสอบได้มีเงื่อนไข

ตาม SLA กรณีเร่งด่วนเสร็จสิ้นภายใน 3 วัน ซึ่งทุกการทดสอบสามารถทำงานได้สำเร็จตาม SLA และในส่วนของการตอบสนองของผู้ใช้งานอยู่ที่ระดับสูง เนื่องจากทุกการทดสอบสามารถทำงานได้สำเร็จตาม SLA ทำให้ผู้ใช้งานมีการตอบสนองผลการใช้งานในแบบสอบถามส่วนมากในระดับดีดังตารางที่ 4.6

ตารางที่ 4.6 ผลสรุปสถิติการติดตั้งระบบเพื่อใช้งานจริงของหน่วยงานในการใช้งานและทดสอบประสิทธิภาพด้วยออโตเมชันเซิร์ฟเวอร์

ขนาดซอฟต์แวร์	**ระยะเวลาสำหรับปรับใช้ (Deploy Lead Time)**	**ความเชื่อมั่น (Reliability)**	**การตอบสนองของผู้ใช้งาน (Customer Responsiveness)**
S	Minutes	High	High
M	Minutes	High	High
L	Minutes	High	High

โดยจากข้อมูลตารางสรุปผลการทดสอบประสิทธิภาพในภาคผนวก ก แสดงให้เห็นถึงประสิทธิภาพของงานที่เพิ่มขึ้นและเป็นไปตามสมมติฐานการวิจัย ทั้งวิธีการดำเนินการวิจัยแบบแมนนวลและออโตเมชันเซิร์ฟเวอร์

ตารางที่ 4.7 แบบสรุปผลการเก็บข้อมูลการทดสอบประสิทธิภาพอย่างย่อจากภาคผนวก ก ลำดับที่ 21 ถึง 28

ลำดับ	Type	Deploy SLA (ชั่วโมง)	Automation Server					Manual Step				
			ระยะเวลาจาก Jenkins Pipeline					ระยะเวลาจาก Serena Business Manager System				
			Push Code	Build	Package	Release	Deploy	Push Code	Build	Package	Release	Deploy
…	…	…	…	…	…	…	…	…	…	…	…	…
21	S	01:00	2s	49s	14s	10s	1s	2s	49s	37s	2s	7h 23m 1s
22	M	02:00	1s	49s	19s	14s	3s	3s	49s	31s	3s	9h 54m 59s
23	M	02:00	1s	49s	19s	9s	3s	1s	49s	46s	5s	1h 42m 15s
24	L	02:00	2s	1m 17s	19s	14s	3s	1s	1m 2s	22s	6s	31h 0m 6s
25	L	01:00	2s	1m 10s	18s	8s	2s	1s	1m	21s	6s	31h 14m 40s
26	M	03:00	2s	1m 10s	20s	8s	2s	2s	1m 27s	23s	11s	3h 27m 33s
27	M	03:00	2s	2m 1s	18s	8s	3s	2s	1m	43s	6s	11h 40m 50s
28	S	03:00	2s	49s	20s	8s	3s	1s	1m	34s	1s	9h 30m 21s
…	…	…	…	…	…	…	…	…	…	…	…	…

ซึ่งจากตารางที่ 4.7 แสดงให้เห็นว่าขั้นตอนการทำงานของวิธีการดำเนินการวิจัยทั้งแบบแมนนวลและออโตเมชันเซิร์ฟเวอร์ใช้ขั้นตอน DevOps ในการทำงานแบบเดียวกัน ต่างเพียงเครื่องมือที่ใช้ในการดำเนินการด้วยมนุษย์ ซอฟต์แวร์ และระบบอัตโนมัติ การนำออโตเมชันเซิร์ฟเวอร์มาช่วยเพื่อประสิทธิภาพของข้อตกลงระดับบริการ สามารถช่วยลดระยะเวลาในขั้นตอนการรอคอยที่เกิดขึ้นจากการทำงานแบบแมนนวลได้ชัดเจนที่สุดในขั้นตอนของการติดตั้ง (Deploy) เนื่องจากในการทำงานแบบแมนนวลจำเป็นต้องอาศัยระยะเวลาในการรอคอยของทีมปฏิบัติการ (Operation) ซึ่งมีการกำหนดรอบของการติดตั้งไว้เป็นรอบตามกำหนดการ เพื่อลดการทำงานของบุคลากรภายในทีมปฏิบัติการที่มีปริมาณมาก และต้องจัดสรรเวลาในการตรวจสอบข้อผิดพลาดที่อาจเกิดขึ้นจากระบบที่ถูกใช้งานอยู่ภายในหน่วยงาน ซึ่งจากตารางที่ 4.7 จะเห็นได้ว่าขนาดของซอฟต์แวร์ไม่ได้ส่งผลต่อความล่าช้าในการให้บริการตาม SLA แต่เกิดจากระยะเวลาในการรอคอยของการส่งต่องานระหว่างทีมของวิธีการแบบแมนนวลที่แม้ว่าจะดำเนินการในขั้นตอนต่างๆได้รวดเร็ว ก็ยังต้องถูกจำกัดด้วยการรอคอยที่ส่งผลต่อความล่าช้าของการให้บริการตาม SLA อันสามารถสังเกตได้จากขั้นตอนการติดตั้ง (Deploy) ในฝั่งของ Manual Step ในลำดับที่ 21 ที่ขนาดซอฟต์แวร์ขนาดเล็ก (S) มีการใช้ระยะเวลาในการติดตั้งอยู่ที่ 7 ชั่วโมง 23 นาที 1 วินาที ที่สามารถอธิบายได้ว่า เมื่อสิ้นสุดกระบวนการเตรียมเผยแพร่ (Release) แล้วและเสร็จสิ้นก่อนรอบการติดตั้งหลายชั่วโมง นักพัฒนาจำเป็นต้องรอรอบการติดตั้งที่ทีมปฏิบัติการกำหนดไว้เพื่อดำเนินการติดตั้งซอฟต์แวร์ที่จัดเตรียมไว้ให้ จึงทำให้ใช้ระยะเวลามากกว่าขนาดกลาง (M) ในลำดับที่ 23 ที่อาจเตรียมการเผยแพร่ช้ากว่า แต่เสร็จสิ้นใกล้กับกำหนดการที่ทีมปฏิบัติการกำหนดเวลาการติดตั้งไว้มากกว่า จึงทำให้ระยะเวลารอคอยการติตตั้งน้อยกว่าลำดับที่ 21 ขนาดเล็ก (S) ได้ แต่หากเป็นวิธีการด้วยออโตเมชันเซิร์ฟเวอร์จะไม่พบปัญหาที่ก่อนให้เกิดความล่าช้าจากการรอคอย เนื่องจากเป็นการทำงานแบบอัตโนมัติที่ทำงานได้โดยไม่ต้องส่งต่องานให้ทีมปฏิบัติการมาทำหน้าที่ในขั้นตอนการติดตั้งจากการอาศัยแนวคิด CI/CD ที่เป็นแนวคิดเพื่อการทำงานแบบอัตโนมัติที่การทำงานแบบแมนนวลไม่สามารถนำมาใช้งานได้ จึงทำให้เกิดความรวดเร็วและเพิ่มประสิทธิภาพการทำงานให้เป็นไปตาม SLA ที่กำหนดไว้

บทที่ 5
สรุปการวิจัย อภิปรายผล และข้อเสนอแนะ

จากการศึกษาและพัฒนาชุดคำสั่งออโตเมชันเซิร์ฟเวอร์ตามวิทยานิพนธ์เรื่อง การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps ตามการผลักดันให้ธุรกิจก้าวสู่ Digital 4.0 เพื่อมุ่งหวังให้ลดบทบาทของมนุษย์ โดยมีการนำเทคโนโลยีเข้ามาใช้หรือแทนที่ในการลดขั้นตอนระยะเวลาการทำงานตามนโยบายของภาครัฐบาลไทย ที่ได้กล่าวไว้ในความเป็นมาและความสำคัญข้างต้นนั้น จึงทำให้ผู้วิจัยได้นำโอกาสดังกล่าวมาใช้เป็นหัวข้องานวิจัยเพื่อศึกษาและพัฒนาการแก้ปัญหาโดยการนำเทคโนโลยีมาสนับสนุนข้อตกลงระดับการบริการทางธุรกิจตามเป้าหมายของหน่วยงานที่ได้กำหนดขึ้นไว้ แต่ยังไม่เกิดระดับอัตราส่วนความสำเร็จเท่าที่ควร ด้วยปัญหาในด้านความล่าช้า ระยะเวลา บุคลากร รวมถึงองค์ประกอบด้านบริบทการทำงานภายในหน่วยงาน จึงมีแนวคิดในการพัฒนาชุดคำสั่งออโตเมชันเซิร์ฟเวอร์เพื่อช่วยลดระยะเวลา ความล่าช้าในการให้บริการของหน่วยงานและมุ่งเน้นด้านการนำเสนอผลงานผลิตภัณฑ์ซอฟต์แวร์ภายในหน่วยงานออกสู่การบริการแก่ผู้ใช้บริการได้รวดเร็วมากขึ้น ซึ่งส่งผลต่อการประเมินศักยภาพการทำงานของบุคลากร รวมถึงภาพลักษณ์ในการแก้ไขปัญหาที่เกิดขึ้นของหน่วยงานให้อยู่ภายใต้ระดับอัตราส่วนที่กำหนดไว้ สามารถสรุปผลและเสนอแนวทางเพื่อศึกษาวิจัยต่อไปได้ในอนาคตดังนี้

1. สรุปผลการวิจัย

1.1 วัตถุประสงค์การวิจัย

1.1.1 เพื่อพัฒนาออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ

1.1.2 เพื่อวิเคราะห์ประสิทธิภาพของออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ

1.1.3 เพื่อประเมินผลการใช้งานของออโตเมชันเซิร์ฟเวอร์ สำหรับการเพิ่มประสิทธิภาพข้อตกลงระดับการบริการตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ

1.2 วิธีดำเนินการวิจัย

ในการวิจัยตามกระบวนการเรียนรู้และศึกษาวิจัยตามวิทยานิพนธ์เรื่อง การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps ผู้วิจัยได้แบ่งวิธีการดำเนินการวิจัยออกเป็น 3 ขั้นตอนสำคัญ ดังนี้

1.2.1 ขั้นตอนการศึกษาทฤษฎี เครื่องมือ และวิธีการพัฒนาออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps โดยผู้วิจัยได้ศึกษาจากเอกสารมาตรฐานและเว็บไซต์ที่เกี่ยวข้องกับการพัฒนาออโตเมชันเซิร์ฟเวอร์ตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง สำหรับขั้นตอนการพัฒนาและปฏิบัติการ อันได้แก่ เอกสารของ Amazon Cloud โดยเป็นเนื้อหาด้านแนวคิดและหลักการทำงานตามรูปแบบของการทำงานสำหรับขั้นตอนการพัฒนาและปฏิบัติการ เอกสารด้านการติดตั้งและใช้งานเครื่องมือที่เกี่ยวข้องกับการใช้ Jenkins รวมถึงเอกสารข้อตกลงระดับบริการทางธุรกิจที่หน่วยงานได้กำหนดแล้วตามมาตรฐาน ISO-27001 : 2013 หมวดระบบบริหารจัดการความมั่นคงปลอดภัยสารสนเทศ (ISMS) และเอกสารที่เกี่ยวข้องกับการทำงานตามแนวคิดกระบวนการบูรณาการ การส่งมอบและปรับใช้แบบต่อเนื่อง เพื่อปรับปรุงให้เหมาะสมกับเทคโนโลยีและบริบทในการทำงานของหน่วยงาน

1.2.2 ขั้นตอนการติดตั้งและพัฒนาชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ผู้วิจัยได้ออกแบบและพัฒนาชุดทำงานออโตเมชันเซิร์ฟเวอร์ด้วยเครื่องมือ Open Source ได้แก่ Jenkins และ GitLab ร่วมกับการทำงานของ MS-Build บนระบบปฏิบัติการ Windows Server 2019 Datacenter Student Version โดยการเขียนชุดคำสั่ง Pipeline Script และการทำ Trigger เพื่อให้การทำงานสามารถทำงานได้อย่างต่อเนื่องตามแนวคิด Continuous Integration และ Continuous Delivery and Deploy ซึ่งชุดคำสั่ง Pipeline Script ร่วมกับคำสั่ง Command Line ของ Windows Server ให้เป็นไปตามแนวคิด CI/CD ของกระบวนการ DevOps ที่ได้กล่าวไว้ในวิธีดำเนินการวิจัย ดังนี้ 1) วางแผน (PLAN) 2) การเขียนโปรแกรม (CODE) 3) การสร้างชุดทำงานโปรแกรม (Build) 4) การทดสอบ (TEST) 5) กระบวนการปล่อยและเวอร์ชัน (RELEASE) 6) การนำไปใช้งานจริง (DEPLOY) และ 7) การปฏิบัติการ (Operate)

1.2.3 ขั้นตอนการทดสอบ ประเมินประสิทธิภาพ และผลการใช้งานชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps การทดสอบและประเมินประสิทธิภาพของชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ทำได้โดยวัดจากจำนวนการทำงานทั้งสิ้น 86 งาน ตามปริมาณจำนวนประชากรที่มีจากกลุ่มบุคลากรที่ประกอบอาชีพด้านเทคโนโลยีสารสนเทศ ที่มีส่วนเกี่ยวข้องกับกระบวนการพัฒนาระบบ ทดสอบระบบ และประเมินการใช้งานเป็นจำนวนคนทั้งสิ้น 61 คน และใช้สถิติในการประเมินผลการใช้งาน

2. อภิปรายผลการวิจัย

การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps โดยใช้พื้นฐานการทำงานบน .NET Framework และระบบปฏิบัติการ Windows Server ควบคู่ไปกับการใช้งาน Jenkins Open Source เพื่อให้เกิดการทำงานแบบต่อเนื่องผ่านชุดคำสั่งการทำงานที่สั่งการให้ทำงานตามบริบทที่ต้องการด้วย Pipeline Script และ Command Line ทำให้เกิดการทำงานในรูปแบบของออโตเมชันเซิร์ฟเวอร์ในกระบวนการสำหรับขั้นตอนการพัฒนาและปฏิบัติการ เพื่อเข้ามามีส่วนช่วยทำให้เกิดอัตราส่วนความสำเร็จตามการสนับสนุนข้อตกลงการบริการทางธุรกิจตามที่หน่วยงานกำหนดไว้

งานวิจัยครั้งนี้พบว่า ค่าเฉลี่ยร้อยละของอัตราส่วนความสำเร็จตามการสนับสนุนข้อตกลงระดับบริการทางธุรกิจจากปริมาณงานทั้งสิ้น 86 งานของการทำงานแบบอัตโนมัติของออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD ของกระบวนการ DevOps และการทำงานแบบแมนนวลตามกระบวนการเดิม คิดเป็นร้อยละ 100 และ 1.16 ตามลำดับ ซึ่งมีปริมาณขนาดของซอฟต์แวร์ที่วัดขนาดตามทฤษฎีโคโคโม (COCOMO) ตามที่หน่วยงานระบุไว้ เรียงตามปริมาณการร่วมทดสอบตามขนาด M, L และ S คิดเป็นอัตราส่วนร้อยละ 52.33, 41.86 และ 5.81 ตามลำดับ ทั้งนี้ในผลการใช้งานจำนวนผู้ประเมินผลการใช้งานทั้งสิ้น 61 คน แบ่งเป็น 3 อันดับแรก ได้แก่ Programmer / Developer / Software Engineer, Tester / Automation Tester / Software QA และ Operator / Operation Deployment ในอัตราส่วนร้อยละ 62.30, 26.20 และ 11.50 ตามลำดับ ซึ่งในผลประเมินในส่วนคำถามเกี่ยวกับการใช้งานและประสิทธิภาพของออโตเมชันเซิร์ฟเวอร์การลดระยะเวลาและขั้นตอนการทำงานจากการใช้งานและการทำงานตามเป้าหมายของ SLA อยู่ที่ระดับ มาก โดยในผลคำตอบของแต่ละหมวดคำถามในแต่ละอันดับแรกที่กล่าวมานั้นมีอัตราส่วนคะแนนโดยรวมอยู่ที่ ร้อยละ 50, 40 และ 10 ตามลำดับ ซึ่งผู้ตอบ

แบบประเมินการใช้งานได้ตอบคำถามปลายเปิดส่วนการแสดงความคิดเห็นและข้อเสนอแนะเพิ่มเติมที่สัมพันธ์กับอัตราส่วนความสำเร็จของประสิทธิภาพการทำงานของออโตเมชันเซิร์ฟเวอร์คือ ออโตชันเซิร์ฟเวอร์สามารถสนับสนุนการทำงานได้อย่างมีประสิทธิภาพ และสามารถทำงานได้ตามระยะเวลาที่กำหนด สอดคล้องกับงานวิจัย 1) Markus Juopperi (2017) ที่ได้สรุปผลการวิจัยเกี่ยวข้องกับการทำงานแบบอัตโนมัติสามารถช่วยให้การทำงานเป็นไปอย่างมีประสิทธิภาพและควบคุมระยะเวลาในการทำงานได้ตามเป้าหมาย 2) S.A.I.B.S. Arachchi, Indika Perera (2018) ที่สอดคล้องในด้านการทำงานอย่างมีประสิทธิภาพของระบบอัตโนมัติว่ากระบวนการทำงานแบบอัตโนมัติสามารถช่วยลดความเสี่ยงในการนำซอฟต์แวร์ไปใช้งานได้อย่างมีประสิทธิภาพ ซึ่งใช้แก้ปัญหาด้านระยะเวลาที่ทำให้เกิดการล่าช้าในกระบวนการพัฒนาและทดสอบระบบแบบเดิมได้อย่างมีนัยสำคัญ และ 3) Kanchan Nair (2019) ที่กล่าวว่าการส่งมอบเพื่อนำไปใช้งานจากการพัฒนาซอฟต์แวร์ด้วยการประมวลผลแบบอัตโนมัติในการคอมไพล์และดำเนินการ ช่วยเพิ่มความรวดเร็วในการทำงานด้วยโครงสร้างการพัฒนาแบบจำลองที่เป็นเครื่องมือสนับสนุนการพัฒนาซอฟต์แวร์ โดยสามารถปรับปรุงประสิทธิภาพของแบบจำลองได้แบบอัตโนมัติทันที และวัดผลจากระยะเวลารวมในการประมวลผลของแต่ละองค์ประกอบแบบจำลอง

ทั้งนี้ในการวิจัยยังพบว่า การนำชุดทำงานออโตเมชันเซิร์ฟเวอร์ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เป็นที่ได้รับความสนใจในกลุ่มผู้ที่ประสบปัญหาด้านความล่าสุดในการทำงานระหว่างทีมมากกว่า 50% และพร้อมที่จะยอมรับการปรับเปลี่ยนและพัฒนาให้เหมาะสมกับบริบทการทำงานให้ดีเพิ่มขึ้น แต่ยังมีบางส่วนที่ยังไม่พร้อมยอมรับการปรับเปลี่ยนเนื่องจากการปรับตัวการรองรับการทำงานของชุดทำงานออโตเมชันเซิร์ฟเวอร์ในบริบทการทำงานของ .NET Framework และปริมาณงานในปัจจุบัน จึงนำไปสู่ข้อจำกัดและการเสนอแนะต่อการทำวิจัยในครั้งต่อไป ซึ่งสอดคล้องกับงานวิจัย Jann Ton (2017) ที่กล่าวไว้ว่า การปรับใช้กับกระบวนการทำงานของ Jenkins เพื่อให้การทำงานแบบอัตโนมัติก่อนส่งมอบแก่ผู้ใช้งาน ช่วยทำให้ผู้ใช้งานและผู้ที่เกี่ยวข้องเกิดความมั่นใจ และเกิดความเชื่อมั่นในประสิทธิภาพการทำงานของซอฟต์แวร์ที่ส่งมอบด้วย อีกทั้งการพัฒนาออโตเมชันเซิร์ฟเวอร์ยังช่วยเพิ่มประสิทธิภาพของอุปกรณ์คอมพิวเตอร์ที่มีอยู่ให้เกิดประสิทธิภาพสูงสุดในการใช้งานและการทำงานของหน่วยงาน เพียงปรับการตั้งค่าของชุดทำงานออโตเมชันเซิร์ฟเวอร์เป็น Application Path และ Solution File ที่ต้องการนำไปใช้งานใน Pipeline Script ที่พัฒนาขึ้น เพื่อลดต้นทุนจากการใช้งานออโตเมชันแบบสำเร็จรูปที่มีค่าใช้จ่ายรายเดือน อาทิ Amazon Web Services หรือ Azure DevOps Server โดยสามารถลดค่าใช้จ่ายอย่างน้อย 86 ผู้ใช้งานจากการอ้างอิงราคาเบื้องต้นของ

Azure DevOps Server ในราคาผู้ใช้งานละ 180 บาท โดยประมาณต่อเดือน รวมเป็นต้นทุนทั้งสิ้น โดยประมาณ 15,480 บาทต่อเดือน หรือประมาณปีละ 185,760 บาท ซึ่งทำให้หน่วยงานมีต้นทุนในการประกอบกิจการที่ลดลงและยังสามารถนำส่วนต่างที่เกิดขึ้นนี้ไปใช้ในการลงทุนเพื่อพัฒนาองค์กรในส่วนอื่นที่ต้องการได้เช่นกัน

3. ข้อจำกัดของการวิจัย

งานวิจัยนี้ได้ทำการทดสอบประสิทธิภาพและประเมินผลการใช้งานจากปริมาณงานทั้งสิ้น 86 งาน โดยแยกเป็นจำนวนคน 61 คน ซึ่งทั้งหมดได้ทดสอบอยู่บนพื้นฐานการทำงานด้วย .NET Framework ที่จำเป็นต้องทำงานบนระบบปฏิบัติการ Windows Server ซึ่งทำให้กลุ่มผู้ใช้งานที่พัฒนาด้วยพื้นฐานการทำงานอื่นไม่สามารถเข้าร่วมการทดสอบประสิทธิภาพและประเมินผลการใช้งานได้ เพราะไม่สามารถทำการเข้าสู่กระบวนการใช้งานจริงได้ ผลการประเมินจึงได้จากกลุ่มบุคคลเพียงกลุ่มเดียว ซึ่งในสังคมการทำงานจริงมีบุคคลการใช้งานมากกว่า 1 กลุ่ม การปรับปรุงให้ยืดหยุ่นและใช้งานร่วมกับกลุ่มบุคคลมากกว่า 1 กลุ่ม จึงจะทำให้ผลการประเมินประสิทธิภาพและการใช้งานถูกต้องและแม่นยำมากขึ้น

4. ข้อเสนอแนะ

4.1 ข้อเสนอแนะที่ได้จากการวิจัยในครั้งนี้

จากการทำงานวิจัยในครั้งนี้ ได้พบข้อควรแก้ไขและปรับปรุงเพื่อให้การวิจัยมีประสิทธิภาพมากขึ้น ดังนี้

1) การทำงานแต่ละงานจะต้องทำการตั้งค่า Virtual Path และ Application Path สำหรับการ Deploy งานแต่ละงานทุกครั้งในครั้งแรกเสมอ ทำให้ต้องอาศัยทีมงานปฏิบัติการทำให้ เนื่องจากมีการจำกัดสิทธิ์การเข้าถึงเครื่องการใช้งานจริง ทำให้เมื่อเกิดการนำไปใช้งานจริงอาจระยะเวลาในการรอคอยการทำงานในครั้งแรก

2) มีกลุ่มการทำงานที่ไม่สามารถเข้าร่วมทดสอบได้จากรูปแบบการทำงานที่อาศัย .NET Framework เพียงอย่างเดียว อาทิ กลุ่มนักพัฒนาใน Mobile Application, Java, ReactJS,

React Native, Go Lang รวมถึง Python เป็นต้น จึงต้องปรับปรุง Pipeline Script ให้สามารถแยกและตรวจสอบการทำงานให้ทำงานได้มากกว่าการทำงานเดียว

3) การทดสอบยังต้องอาศัยการทดสอบแบบแมนวล โดยอาศัยทีมการทดสอบทำหน้าที่ในการทดสอบ เนื่องจากด้วยระยะเวลาที่จำกัดในงานวิจัยจึงไม่สามารถพัฒนาชุดการทำงานการทดสอบแบบอัตโนมัติของแต่ละงานได้ทัน การเตรียมพร้อมของแต่ละหน่วยงานในการทำให้เกิดการทดสอบแบบอัตโนมัติของแต่ละงานจึงเป็นสิ่งที่จำเป็นหากต้องการให้เกิดระบบอัตโนมัติ 100 เปอร์เซ็นต์

4.2 ข้อเสนอแนะในการวิจัยครั้งต่อไป

จากข้อเสนอแนะในงานวิจัยในครั้งนี้ สามารถนำมาเป็นข้อเสนอแนะในการวิจัยครั้งต่อไป เพื่อให้เกิดงานวิจัยในกลุ่มการวิจัยที่เกี่ยวข้องกับการพัฒนาออโตเมชันเซิร์ฟเวอร์ในอนาคต ดังนี้

1) Pipeline Script ควรมีการรับค่าตัวแปรสำหรับการสร้าง Virtual Path และ Application Path ให้เกิดการตรวจสอบและใช้คำสั่ง Command Line บน Windows Server เพื่อสร้าง Virtual Path หรือ Application Path ให้อัตโนมัติ เพื่อไม่ให้เกิดความล่าช้าจากการส่งต่องานระหว่างทีมปฏิบัติการ

2) เปลี่ยนกลุ่มการทำงานจาก .NET Framework ให้รองรับการทำงานของ .NET Core หรือกำหนดเงื่อนไขการ Build เมื่อไม่ใช่ Project ของ .NET Framework ให้เป็นการ Build ด้วยตัว Build อื่นที่ไม่ใช่ MS-Build

3) การทดสอบแบบอัตโนมัติต้องอาศัยทีมพัฒนาการทดสอบแบบอัตโนมัติ เช่น Robot Framework ซึ่งเป็นกรอบการทำงานแบบ Open Source สำหรับการพัฒนากระบวนการทดสอบแบบอัตโนมัติ มาช่วยในการพัฒนาชุดของกรณีทดสอบ (Test Script), กรณีทดสอบ (Test Case) และลำดับขั้นของกรณีทดสอบ (Test Scenario) ต่างๆ ให้เหมาะสมกับงานที่ต้องการใช้งานกับชุดทำงานออโตเมชันเซิร์ฟเวอร์ เพิ่ม Stage การทดสอบใน Pipeline Script ให้มีการ Trigger ไปยัง API ของตัวทดสอบอัตโนมัติให้ทำหน้าที่ทดสอบอัตโนมัติ เพื่อให้เกิดความรวดเร็วและเป็นระบบอัตโนมัติ 100 เปอร์เซ็นต์

บรรณานุกรม

บรรณานุกรม

กองวิเคราะห์และตรวจสอบทรัพยากรธรณี กรมทรัพยากรธรณี. (2562). ข้อตกลงระดับการให้บริการ. *กรมทรัพยากรธรณี*. สืบค้นจาก http://www.dmr.go.th/download/ข้อตกลงระดับการให้บริการ .pdf

ธนาวิชญ์ จินดาประดิษฐ์. (2561). *แนวทางการพัฒนาระบบงานและกระบวนการทำงานสู่การเป็นองค์กรดิจิทัล.* กรุงเทพมหานคร: สำนักงานคณะกรรมการพัฒนาระบบราชการ.

บริษัท ที-เน็ต จำกัด. (2563). มาตรฐาน ISO-27001 : 2013. *RTNA*. สืบค้นจาก http://www.rtna.ac.th/download/27001-2013.pdf

บริษัท สุมิพล คอร์ปอเรชั่น จำกัด. (2562, 24 กันยายน). แนวโน้มการใช้ซอฟท์แวร์และระบบออโตเมชั่น. *Sumipol.* สืบค้นจาก https://www.sumipol.com/knowledge/ซอฟท์แวร์ออโตเมชั่น/

บริษัท แอดวานซ์ อินโฟร์ เซอร์วิส จำกัด (มหาชน). (2560). *รายงานประจำปี และ รายงานความยั่งยืน ประจำปี 2560.* บริษัท แอดวานซ์ อินโฟร์ เซอร์วิส จำกัด (มหาชน): กรุงเทพฯ.

ปิยวรรณ ปนิทานเต. (2560). ระบบการผลิตอัจฉริยะเป็นอย่างไร ในยุคอุตสาหกรรม 4.0. *MTEC*, น. 41-46.

ปิยะวัฒน์ ศรีธรรม. (2561). แขนกลจับชิ้นงานจากเครื่องคัดแยกวัสดุอัตโนมัติตามสายพาน. *FEAT Journal : Farm Engineering and Automation Technology Journal* (น. 19-27). ขอนแก่น: มหาวิทยาลัยขอนแก่น.

แผนกนโยบายและแผนฝ่ายยุทธศาสตร์อุตสาหกรรม สถาบันไฟฟ้าและอิเล็กทรอนิกส์. (2560). *แนวทางการขับเคลื่อนอุตสาหกรรมหุ่นยนต์และระบบอัตโนมัติของไทย.* กรุงเทพมหานคร: สถาบันไฟฟ้าและอิเล็กทรอนิกส์ (Electrical and Electronics Institute).

บรรณานุกรม (ต่อ)

พรจิต ประทุมสุวรรณ. (2561). *หุ่นยนต์และระบบอัตโนมัติในอุตสาหกรรม 4.0.* กรุงเทพมหานคร: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ.

พรภวิษย์ สุขโภคี. (2562). *การนำ ISMS มาประยุกต์ใช้กับการให้บริการของศูนย์ปฏิบัติการความมั่นคงสารสนเทศ.* กรุงเทพฯ: คณะวิทยาการและเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีมหานคร.

มหศักดิ์ เกตุฉ่ำ, ดร. (2562). *CMMI (Capability Maturity Model Integration).* กรุงเทพฯ: KMUTNB - Faculty of Information.

สมศักดิ์ ขาวสุวรรณ, นาวาอากาศเอก. (2561). *ข้อตกลงระดับการให้บริการร่วมกันระหว่างหน่วยงานองค์กรดิจิทัล.* กระทรวงดิจิทัลเศรษฐกิจและสังคม. สืบค้นจาก https://dga.or.th/upload/download/file_f855d91bd28bfba036c7d7959b5ba718.pdf

วรินทร์วงษ์มณี, ธรรมรัตน์ ปัญญาภรณ์ประสาท. (2559). การพัฒนาระบบ IT Support ด้วยหลักการลีนเพื่อเพิ่มประสิทธิภาพการบริการกรณีศึกษาบริษัท D Thailand จำกัด. *การประชุมสัมมนาเชิงวิชาการด้านการจัดการโลจิสติกส์และโซ่อุปทาน ครั้งที่ 16.* กรุงเทพมหานคร: มหาวิทยาลัยหอการค้าไทย.

สถาบันยานยนต์. (2559). บทที่ 1 การหาความต้องการในการใช้ระบบอัตโนมัติ (Automation Requirement). ใน แผนกพัฒนาผู้ประกอบการ สถาบันยานยนต์, *การประยุกต์ใช้ระบบอัตโนมัติในการผลิตชิ้นส่วนยานยนต์* (น. 1-7). กรุงเทพมหานคร: สถาบันยานยนต์ โดย กระทรวงอุตสาหกรรม.

สนิธพรรณ จิตตั้งสมบูรณ์. (2559). *ปัจจัยที่ส่งผลต่อความตั้งใจในการใช้งานแนวคิดการพัฒนาซอฟต์แวร์รูปแบบ DevOps.* กรุงเทพมหานคร: คณะพาณิชยศาสตร์และการบัญชี มหาวิทยาลัยธรรมศาสตร์.

บรรณานุกรม (ต่อ)

สมลักษณ์ วรรณฤมล กีเยลาโรว่า, ประภัสสร ประดุจพงษ์เพชร, และ โชคนิธิ นาคเมธี. (2561). การพัฒนาระบบออกแบบอัตโนมัติเพื่อช่วยในการออกแบบเครื่องประดับแหวนตามความชื่นชอบของนักท่องเที่ยวชาวจีน. *Naresuan University Journal: Science and Technology* (น. 107-123). พิษณุโลก: มหาวิทยาลัยนเรศวร.

สำนักงานคณะกรรมการดิจิทัลเพื่อเศรษฐกิจและสังคมแห่งชาติ. (2561). นโยบายและแผนระดับชาติว่าด้วยการพัฒนาดิจิทัลเพื่อเศรษฐกิจและสังคม. *ราชกิจจานุเบกษา*, น. 27-30.

สำนักงานปลัดกระทรวงศึกษาธิการ. (2561, 25 กันยายน). *ยุค Digital 4.0 เมื่อโลกขับเคลื่อนด้วยเทคโนโลยี (ภาพ)*. สำนักงานปลัดกระทรวงศึกษาธิการ. สืบค้นจาก http://www.ops.moe.go.th/ops2017/สาระน่ารู้/2876-ยุค-digital-4-0-เมื่อโลกขับเคลื่อนด้วยเทคโนโลยี-25-ก-ย-2561

สำนักบริการคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์. (2543). *โครงสร้างองค์กรกับระบบสารสนเทศ.* KU Electric Magazine. สืบค้นจาก https://www.ku.ac.th/e-magazine/september43/information/

องค์การพิพิธภัณฑ์วิทยาศาสตร์แห่งชาติ (อพวช.). (2561). *ข้อตกลงระดับการให้บริการ*. องค์การพิพิธภัณฑ์วิทยาศาสตร์แห่งชาติ (อพวช.). สืบค้นจาก http://www.thai-sciencemuseum.com/Portals/internal-control/Manual/Training/2561/เอกสารประกอบการอบรมเชิงปฏิบัติการ-ข้อตกลงระดับการให้บริการ.pdf

องอาจ วรฤทธิ์โสภณ. (2559). *การจัดกระบวนการให้บริการด้าน IT ให้มีมาตรฐานการทำงานโดยใช้หลักการ ITIL V.3 (กรณีศึกษา: บริษัท ไทยโพลิเมอร์ ซัพพลาย จำกัด).* กรุงเทพมหานคร: คณะวิทยาการและเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีมหานคร.

บรรณานุกรม (ต่อ)

AIS. (2563). ข้อความระบบแจ้งการติดตามปัญหาและตรวจสอบการใช้งาน. กรุงเทพมหานคร.

Aman, A. (2019). *Software Engineering | COCOMO II Model.* Geeks for Geeks. สืบค้นจาก https://www.geeksforgeeks.org/software-engineering-cocomo-ii-model/

Amazon Web Services, Inc. (2562). *DevOps คืออะไร*. AWS Site. สืบค้นจาก https://aws.amazon.com/th/devops/what-is-devops/

Arachchi & Perera. (2018). Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management. *MERCon 2018.* Moratuwa, Sri Lanka: Department of Computer Science and Engineering, University of Moratuwa. สืบค้นจาก https://www.researchgate.net/publication/326406017_Continuous_Integration_and_Continuous_Delivery_Pipeline_Automation_for_Agile_Software_Project_Management

Bouman, Abran & April. (2000). Software Maintenance in a Service Level Agreement: Controlling the Customers Expectations. *FESMA-AEMES Software Measurement Conference 2000.* Madrid, Spain: FESMA-AEMES.

BSA | The Software Alliance. (2562). *มาตรฐานอุตสาหกรรม*. BSA | The Software Alliance. สืบค้นจาก https://ww2.bsa.org/anti-piracy/what-is-sam/industry-standards?sc_lang=th-TH

Clark & Virga. (2018). *Continuous Integration and Continuous Delivery: Beyond the Conveyor Belt Mentality*. SIRIUS Edge. สืบค้นจาก https://edge.siriuscom.com/digital/continuous-integration-and-continuous-delivery-beyond-the-conveyor-belt-mentality

DIP-SME ACAADEMY. (2561). *ธุรกิจดิจิตอลเป็นธุรกิจของทุกคน*. DIP-SME ACAADEMY. สืบค้นจาก https://dip-sme-academy.com/knowleagehub/article/150-ธุรกิจดิจิตอลเป็นธุรกิจของทุกคน

Effio, V.H. (2019). *Setting up a CI/CD pipeline in Azure DevOps for Web App + Sql Server and Deploy to Azure*. Medium. สืบค้นจาก https://medium.com/pipelinespace/setting-up-a-ci-cd-pipeline-in-azure-devops-for-web-app-sql-server-and-deploy-to-azure-7b9a92052bd6

บรรณานุกรม (ต่อ)

Faustino, P.C.J. (2018). DevOps Practices in Incident Management Process. (Master's thesis). University Institute of Lisbon, Department of Information Science and Technology.

Hakli, A. (2016). Implementation of Continuous Delivery Systems. (Master's thesis). Tampere University of Technology, Faculty Council of the Faculty of Computer and Electrical Engineering.

iFew in Technology. (2561). *เริ่มต้นทำ CI/CD – Automation Deployment ด้วย Git และ Jenkins*. myIFew.com. สืบค้นจาก https://myifew.com/4072/automation-deploy-git-bitbucket-jenkins/

Lan, S. (2011). *Software Engineering* (พิมพ์ครั้งที่ 9). Pearson Education Asia.

LearnIT. (2563). *DATA CENTER AUTOMATION*. LearnIT. สืบค้นจาก https://www.leanit.com.cy/it-operations-management-itom/data-center-automation/

Leite, Aguiar, Kon, Milojicic & Paulo Meirelles. (2562). *A Survey of DevOps Concepts and Challenges.* Research Gate.

Lopez, J. (2014). *Digital Business is Everyone's Business*. Forbes. สืบค้นจาก https://www.forbes.com/sites/gartnergroup/2014/05/07/digital-business-is-everyones-business/#7d2e2437f82b

Microsoft. (2016). *Automation Servers*. Microsoft. สืบค้นจาก https://docs.microsoft.com/en-us/cpp/mfc/automation-servers?view=vs-2019

MindPHP. (2562). *CMMI คือ อะไร มาตรฐานกระบวนการในการพัฒนางาน*. MindPHP.COM. สืบค้นจาก https://mindphp.com/คู่มือ/73-คืออะไร/4113-what-is-cmmi.html

MindPHP.COM. (2560). *KPI คืออะไร การตั้ง KPI อย่างไรให้เหมาะกับองค์กร การพัฒนา Software*. MindPHP.COM. สืบค้นจาก https://mindphp.com/บทความ/31-ความรู้ทั่วไป/3217-kpi.html

บรรณานุกรม (ต่อ)

Motadata ITSM. (2562). *การทำความเข้าใจข้อตกลงระดับบริการ (SLA)*. Motadata ITSM. สืบค้นจาก https://www.motadata.com/th/blog/types-of-service-level-agreement/

NOSTRA. (2563). *มาตรฐานการให้บริการ NOSTRA Map Online (SLA)*. NOSTRA Map APIs. สืบค้นจาก https://developer.nostramap.com/developer/V1/serviceLevelAgreement.html

Paloposki, A. (2018). Enabling Continuous Integration through deployment automation Case Study: Property transaction system ofFinnish National Land Survey. (Master's thesis). Aalto University, School of Electrical Engineering.

Pathania, N. (2017). *Learning Continuous Integration with Jenkins* (พิมพ์ครั้งที่ 2). BIRMINGHAM - MUMBAI: Packt Publishing Ltd.

Qoppa Software. (2563). *PDF Automation Server – Workflow Module*. Qoppa Software. สืบค้นจาก https://www.qoppa.com/wp-content/themes/rttheme17/images/PAS.gif

Raza, M. (2020). Service Level Agreement (SLA) Examples and Template. O*ptimizing for success : Metrics for enterprise IT*. United States : BMC Software, Inc.

Saknimitwong, P. (2560). *Learn DevOps ตอนที่ 1 : จุดเริ่มต้นของการเปลี่ยนแปลง*. Medium. สืบค้นจาก https://medium.com/@pariwat_s/learn-devops-ตอนที่-1-จุดเริ่มต้นของการเปลี่ยนแปลง-e95640a39a10

Saknimitwong, P. (2560). *Learn DevOps ตอนที่ 2 : DevOps คืออะไร ?*. Medium. สืบค้นจาก https://medium.com/@pariwat_s/learn-devops-ตอนที่-2-devops-คืออะไร-18ac48d73625

Sakulsubwatthana, W. (2561). *5 คุณสมบัติที่สำคัญของ DevOps*. Medium. สืบค้นจาก https://medium.com/linedevth/5-คุณลักษณะสำคัญของ-devops-1b4df97ed70d

Sangakong, A. (2563). *What is DevOps?*. สืบค้นจาก https://welovebug.com/what-is-devops-e9596f144738

บรรณานุกรม (ต่อ)

Santos-Pereira, Rijo & Lourenço. (2014). Service Level Agreement of Information and Communication Technologies inPortugueseHospitals. *CENTERIS 2014 - Conference on ENTERprise Information Systems / ProjMAN 2014 - International Conference on Project MANagement/ HCIST 2014 - International Conference on Health and Social Care Information Systems and Technologies.* Elsevier Ltd.

Singh, J. (2018). *DevOps without DevOps tools*. Medium. สืบค้นจาก https://medium.com/faun/devops-without-devops-tools-3f1deb451b1c

Swapan, A. (2020). *Software Engineering | COCOMO Model.* Geeks for Geeks. สืบค้นจาก https://www.geeksforgeeks.org/?p=193526

The Chartered Institute of Purchasing & Supply. (2009). How to prepare Service Level Agreements. *CIPS KNOWLEDGE WORKS, Knowledge How To*, น.15.

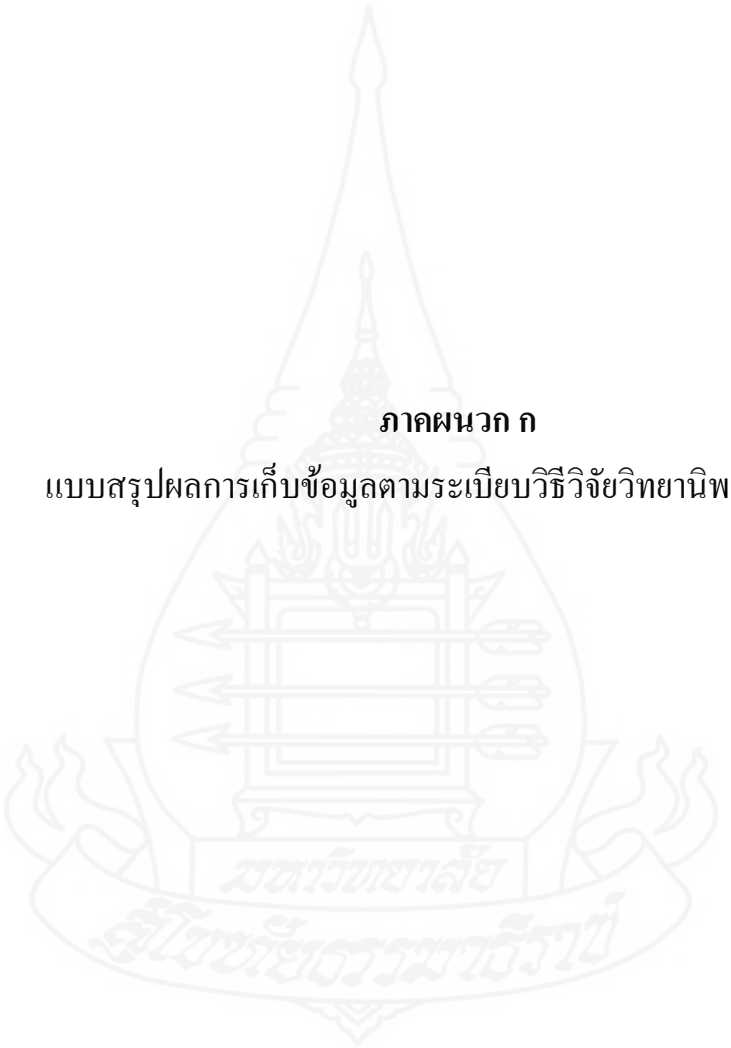
The Office of the University Press และ มหาวิทยาลัยสุโขทัยธรรมธิราช. (2561). *คู่มือข้อตกลงระดับการให้บริการการผลิตเอกสารการสอนชุดวิชา*. มหาวิทยาลัยสุโขทัยธรรมาธิราช. สืบค้นจาก https://www.stou.ac.th/Offices/oup/km web/images/PDF/27-11-61 คู่มือ SLA สำนักพิมพ์ 26-11-2561.pdf

Tsalolikhin, A. (2018). *INTRODUCTION TO CI/CD: CONCEPTSAND TERMINOLOGY*. สืบค้นจาก http://www.verticalsysadmin.com/cicd-intro.pdf

ภาคผนวก

ภาคผนวก ก

แบบสรุปผลการเก็บข้อมูลตามระเบียบวิธีวิจัยวิทยานิพนธ์



แบบสรุปผลการเก็บข้อมูลตามระเบียบวิธีวิจัยวิทยานิพนธ์

เรื่อง การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps

ลำดับ	Type	Deploy SLA (ชั่วโมง)	Automation Server							Manual Step						
			ระยะเวลาจาก Jenkins Pipeline					Total Time (ชั่วโมง)	Deploy SLA Complete	ระยะเวลาจาก Serena Bussiness Manager System					Total Time (ชั่วโมง)	Deploy SLA Complete
			Push Code	Build	Package	Release	Deploy			Push Code	Build	Package	Release	Deploy		
1	L	03:00	1s	51s	14s	8s	2s	00:02	Yes	1s	45s	40s	2s	22h 51m 26s	23:28	No
2	L	02:00	1s	50s	13s	8s	1s	00:02	Yes	1s	50s	52s	2s	5h 48m 22s	06:24	No
3	L	03:00	2s	1m 2s	18s	8s	2s	00:03	Yes	1s	59s	10s	2s	4h 50m 44s	05:27	No
4	L	03:00	2s	1m 11s	19s	8s	1s	00:03	Yes	1s	1m 8s	14s	3s	37h 35m 2s	38:01	No
5	L	02:00	1s	49s	14s	14s	2s	00:02	Yes	1s	48s	42s	1s	32h 1m 15s	32:05	No
6	L	02:00	2s	1m 37s	17s	14s	2s	00:04	Yes	2s	1m 31s	26s	7s	7h 28m 43s	07:51	No
7	L	03:00	1s	49s	14s	7s	2s	00:02	Yes	2s	49s	37s	4s	32h 21m 7s	32:38	No
8	L	01:00	2s	1m 7s	16s	7s	2s	00:03	Yes	2s	1m 7s	13s	3s	29h 42m 38s	30:13	No
9	M	01:00	3s	1m	16s	9s	2s	00:01	Yes	2s	1m 11s	44s	11s	31h 21m 23s	31:39	No
10	L	01:00	2s	1m 11s	18s	15s	1s	00:03	Yes	2s	1m 11s	53s	5s	27h 51m 12s	28:29	No
11	L	01:00	1s	49s	14s	8s	1s	00:02	Yes	3s	45s	57s	16s	8h 10m 20s	08:21	No
12	M	03:00	1s	1m	20s	6s	3s	00:03	Yes	2s	54s	45s	3s	2h 26m 22s	02:47	Yes
13	M	02:00	1s	1m 2s	18s	7s	1s	00:02	Yes	1s	1m	31s	4s	7h 42m 42s	08:14	No
14	M	01:00	2s	1m 1s	16s	7s	1s	00:02	Yes	3s	1m 1s	29s	4s	11h 23m 56s	11:43	No
15	L	02:00	3s	1m 11s	17s	9s	1s	00:03	Yes	3s	1m 2s	25s	14s	7h 8m 6s	07:16	No
16	M	02:00	3s	49s	16s	9s	2s	00:02	Yes	3s	49s	50s	4s	8h 51m 58s	09:30	No
17	M	03:00	3s	49s	13s	16s	1s	00:02	Yes	3s	49s	14s	3s	12h 45m 36s	13:18	No
18	L	02:00	2s	50s	12s	18s	2s	00:02	Yes	2s	39s	40s	1s	32h 3m 29s	32:08	No
19	L	01:00	3s	1m 7s	18s	18s	2s	00:03	Yes	3s	51s	46s	2s	37h 56m 54s	38:38	No
20	M	02:00	2s	1m	18s	15s	2s	00:03	Yes	2s	1m	11s	1s	13h 53m 5s	14:31	No
21	S	01:00	2s	49s	14s	10s	1s	00:02	Yes	2s	49s	37s	2s	7h 23m 1s	07:41	No
22	M	02:00	1s	49s	19s	14s	3s	00:02	Yes	3s	49s	31s	3s	9h 54m 59s	10:34	No
23	M	02:00	1s	49s	19s	9s	3s	00:02	Yes	1s	49s	46s	5s	1h 42m 15s	02:13	No
24	L	02:00	2s	1m 17s	19s	14s	3s	00:03	Yes	1s	1m 2s	22s	6s	31h 0m 6s	31:03	No
25	L	01:00	2s	1m 10s	18s	8s	2s	00:03	Yes	1s	1m	21s	6s	31h 14m 40s	31:27	No
26	M	03:00	2s	1m 10s	20s	8s	2s	00:03	Yes	2s	1m 27s	23s	11s	3h 27m 33s	03:49	No
27	M	03:00	2s	2m 1s	18s	8s	3s	00:04	Yes	2s	1m	43s	6s	11h 40m 50s	12:11	No

แบบสรุปผลการเก็บข้อมูลตามระเบียบวิธีวิจัยวิทยานิพนธ์

เรื่อง การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps

ลำดับ	Type	Deploy SLA (ชั่วโมง)	Automation Server							Manual Step						
			ระยะเวลาจาก Jenkins Pipeline					Total Time (ชั่วโมง)	Deploy SLA Complete	ระยะเวลาจาก Serena Bussiness Manager System					Total Time (ชั่วโมง)	Deploy SLA Complete
			Push Code	Build	Package	Release	Deploy			Push Code	Build	Package	Release	Deploy		
28	S	03:00	2s	49s	20s	8s	3s	00:02	Yes	1s	1m	34s	1s	9h 30m 21s	09:53	No
29	M	02:00	1s	49s	19s	8s	3s	00:02	Yes	1s	49s	11s	5s	34h 23m 48s	34:42	No
30	L	02:00	3s	57s	19s	14s	3s	00:03	Yes	3s	1m	55s	6s	6h 51m 32s	07:29	No
31	L	02:00	1s	49s	14s	14s	3s	00:02	Yes	1s	55s	22s	8s	6h 47m 19s	07:21	No
32	M	02:00	2s	58s	19s	10s	1s	00:03	Yes	1s	54s	5s	4s	35h 57m 31s	36:38	No
33	M	02:00	1s	1m 2s	14s	9s	2s	00:02	Yes	1s	59s	25s	9s	8h 50m 34s	09:27	No
34	S	03:00	1s	1m 11s	13s	8s	1s	00:03	Yes	1s	1m 3s	57s	1s	33h 26m 7s	33:47	No
35	L	03:00	1s	1m 11s	14s	14s	1s	00:03	Yes	1s	1m 4s	35s	18s	31h 1m 54s	31:06	No
36	L	03:00	1s	1m 11s	11s	14s	3s	00:03	Yes	1s	1m 51s	9s	12s	33h 48m 40s	34:25	No
37	L	03:00	3s	59s	12s	17s	3s	00:03	Yes	1s	49s	51s	20s	6h 56m 8s	07:37	No
38	L	03:00	1s	1m	14s	17s	3s	00:03	Yes	1s	1m 21s	30s	6s	28h 10m 8s	28:20	No
39	L	03:00	2s	49s	15s	18s	3s	00:02	Yes	2s	49s	58s	6s	8h 52m 23s	03:31	No
40	L	03:00	2s	1m	15s	17s	3s	00:03	Yes	2s	1m	34s	4s	30h 1m 37s	30:04	No
41	M	01:00	2s	1m	17s	19s	6s	00:01	Yes	2s	39s	22s	5s	30h 19m 19s	30:34	No
42	M	02:00	2s	1m 7s	18s	14s	2s	00:03	Yes	2s	1m 7s	15s	8s	7h 4m 51s	07:11	No
43	L	01:00	2s	57s	18s	18s	2s	00:03	Yes	2s	56s	39s	16s	5h 6m 28s	05:14	No
44	M	01:00	3s	49s	18	14s	1s	00:02	Yes	2s	49s	12s	5s	6h 39m 28s	07:08	No
45	M	02:00	2s	49s	19s	14s	1s	00:02	Yes	2s	49s	45s	3s	11h 2m 9s	11:06	No
46	M	02:00	1s	57s	15s	14s	1s	00:03	Yes	1s	49s	46s	9s	31h 19m 48s	31:36	No
47	M	02:00	1s	57s	13s	16s	3s	00:03	Yes	1s	49s	9s	7s	3h 53m 23s	04:31	No
48	M	02:00	1s	57s	14s	16s	3s	00:03	Yes	1s	51s	29s	9s	7h 11m 43s	07:22	No
49	M	02:00	1s	1m	14s	14s	4s	00:03	Yes	1s	1m	49s	7s	30h 51m 6s	31:28	No
50	M	02:00	1s	58s	14s	16s	2s	00:02	Yes	1s	55s	21s	2s	32h 9m 16s	32:18	No
51	S	03:00	1s	48s	15s	9s	2s	00:02	Yes	1s	41s	11s	5s	31h 49m 49s	32:25	No
52	M	01:00	1s	1m 18s	17s	10s	3s	00:03	Yes	1s	56s	27s	6s	10h 0m 24s	10:03	No
53	M	03:00	1s	1m 17s	16s	9s	3s	00:03	Yes	1s	49s	6s	7s	9h 47m 10s	10:20	No
54	M	03:00	1s	1m 17s	17s	9s	3s	00:03	Yes	1s	51s	7s	8s	25h 17m 1s	25:30	No

แบบสรุปผลการเก็บข้อมูลตามระเบียบวิธีวิจัยวิทยานิพนธ์

เรื่อง การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps

ลำดับ	Type	Deploy SLA (ชั่วโมง)	Automation Server							Manual Step						
			ระยะเวลาจาก Jenkins Pipeline					Total Time (ชั่วโมง)	Deploy SLA Complete	ระยะเวลาจาก Serena Bussiness Manager System					Total Time (ชั่วโมง)	Deploy SLA Complete
			Push Code	Build	Package	Release	Deploy			Push Code	Build	Package	Release	Deploy		
55	M	01:00	2s	1m	17s	7s	3s	00:02	Yes	2s	59s	47s	5s	30h 34m 15s	31:00	No
56	M	03:00	2s	1m 11s	17s	14s	3s	00:01	Yes	2s	58s	45s	9s	33h 54m 36s	34:34	No
57	M	03:00	2s	49s	15s	14s	3s	00:02	Yes	2s	52s	34s	1s	33h 30m 6s	33:53	No
58	M	01:00	2s	52s	15s	15s	3s	00:02	Yes	2s	52s	47s	5s	30h 9m 7s	30:18	No
59	M	02:00	1s	49s	16s	14s	4s	00:02	Yes	1s	52s	16s	2s	11h 34m 7s	11:59	No
60	M	01:00	3s	52s	13s	14s	4s	00:02	Yes	1s	52s	55s	4s	7h 43m 22s	08:15	No
61	L	01:00	3s	52s	19s	16s	4s	00:03	Yes	2s	59s	44s	8s	12h 20m 19s	12:37	No
62	L	02:00	3s	50s	20s	18s	4s	00:03	Yes	2s	1m 2s	25s	8s	34h 33m 31s	34:59	No
63	L	02:00	2s	1m 3s	19s	18s	1s	00:03	Yes	2s	1m	44s	3s	12h 10m 12s	12:20	No
64	M	02:00	1s	1m	18s	14s	1s	00:03	Yes	1s	1m	43s	9s	34h 44m 57s	35:18	No
65	S	03:00	2s	1m 3s	15s	14s	1s	00:03	Yes	2s	52s	42s	17s	33h 21m 41s	33:39	No
66	M	02:00	2s	59s	16s	15s	1s	00:03	Yes	2s	59s	13s	7s	29h 31m 40s	29:55	No
67	M	01:00	2s	1m	16s	14s	2s	00:01	Yes	2s	1m	23s	13s	7h 0m 34s	07:04	No
68	L	02:00	1s	2m	19s	14s	2s	00:01	Yes	1s	1m 12s	30s	4s	31h 18m 21s	31:34	No
69	L	02:00	1s	48s	18s	14s	2s	00:02	Yes	1s	49s	29s	6s	37h 12m 17s	37:23	No
70	M	01:00	1s	48s	17s	14s	3s	00:02	Yes	1s	48s	16s	12s	36h 48m 45s	37:23	No
71	M	02:00	1s	45s	15s	14s	3s	00:02	Yes	1s	45s	42s	7s	6h 59m 16s	07:01	No
72	M	01:00	1s	1m 1s	17s	15s	3s	00:03	Yes	1s	45s	22s	12s	30h 13m 47s	30:25	No
73	M	01:00	1s	55s	17s	14s	2s	00:02	Yes	1s	56s	21s	2s	29h 13m 17s	29:24	No
74	M	02:00	1s	56s	11s	14s	2s	00:02	Yes	1s	56s	28s	2s	11h 49m 49s	12:25	No
75	L	02:00	2s	1m 7s	15s	15s	2s	00:03	Yes	2s	57s	17s	9s	28h 17m 38s	28:32	No
76	M	01:00	1s	54s	14s	8s	2s	00:02	Yes	2s	55s	56s	13s	35h 27m 14s	35:49	No
77	M	02:00	2s	54s	16s	14s	2s	00:02	Yes	2s	55s	11s	12s	34h 54m 46s	35:34	No
78	M	01:00	2s	54s	14s	10s	2s	00:02	Yes	2s	1m	42s	13s	32h 33m 39s	32:59	No
79	L	01:00	1s	59s	14s	7s	2s	00:02	Yes	2s	1m	52s	9s	5h 56m 48s	06:38	No
80	L	02:00	3s	59s	16s	14s	2s	00:03	Yes	2s	45s	34s	6s	30h 41m 30s	31:12	No
81	L	02:00	1s	1m 1s	17s	14s	2s	00:03	Yes	1s	1m 17s	52s	19s	13h 46m 50s	14:22	No

แบบสรุปผลการเก็บข้อมูลตามระเบียบวิธีวิจัยวิทยานิพนธ์

เรื่อง การเพิ่มประสิทธิภาพข้อตกลงระดับการบริการด้วยออโตเมชันเซิร์ฟเวอร์ตามแนวคิด CI/CD สำหรับ DevOps

ลำดับ	Type	Deploy SLA (ชั่วโมง)	Automation Server							Manual Step						
			ระยะเวลาจาก Jenkins Pipeline					Total Time (ชั่วโมง)	Deploy SLA Complete	ระยะเวลาจาก Serena Bussiness Manager System					Total Time (ชั่วโมง)	Deploy SLA Complete
			Push Code	Build	Package	Release	Deploy			Push Code	Build	Package	Release	Deploy		
82	L	02:00	2s	1m	17s	14s	2s	00:03	Yes	2s	1m	58s	5s	3h 44m 15s	04:17	No
83	L	02:00	2s	1m 54s	19s	15s	1s	00:04	Yes	2s	1m 45s	14s	7s	35h 58m 14s	36:01	No
84	L	02:00	2s	1m 2s	17s	15s	1s	00:03	Yes	2s	58s	15s	5s	36h 47m 14s	37:21	No
85	M	02:00	2s	52s	17s	7s	2s	00:02	Yes	2s	55s	28s	5s	35h 45m 48s	36:19	No
86	M	02:00	3s	58s	17s	8s	1s	00:02	Yes	3s	58s	59s	9s	31h 30m 15s	31:54	No

รวมกลุ่มตัวอย่าง	**86**
สำเร็จตาม SLA	**86**
ไม่สำเร็จตาม SLA	**0**
เปอร์เซ็นต์ความสำเร็จ	**100.00**
เปอร์เซ็นต์ไม่สำเร็จ	**0.00**

รวมกลุ่มตัวอย่าง	**86**
สำเร็จตาม SLA	**1**
ไม่สำเร็จตาม SLA	**85**
เปอร์เซ็นต์ความสำเร็จ	**1.16**
เปอร์เซ็นต์ไม่สำเร็จ	**98.84**

หมายเหตุ L = Large ms = Millisecond h = Hour Yes = Complete

M = Medium s = Second d = Day No = Incomplete

S = Small m = Minute

** Type อาศัยการวัดขนาดซอฟแวร์ด้วยปัจจัยตามขนาดที่ระบุใน Request Document ของระบบ ซึ่งยึดตามทฤษฎี COCOMO

ภาคผนวก ข

แบบประเมินการใช้งาน Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps เพื่อรองรับการสนับสนุน Service Level Agreements ทางธุรกิจ

ลำดับ	รายการสอบถาม	ระดับคะแนน				
		5 – มากที่สุด	4 – มาก	3 – ปานกลาง	2 – น้อย	1 – น้อยที่สุด
1	**การใช้งานและประสิทธิภาพของ Automation Server**	-	-	-	-	-
	1.1 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถทำงานได้ตามขั้นตอนที่กำหนดไว้					
	1.2 Automation Server ภายใต้แนวคิด CI/CD สามารถนำไปใช้งานในการทำงานได้จริงภายใต้กระบวนการ DevOpv					
	1.3 Automation Server ภายใต้แนวคิด CI/CD สามารถใช้งานทดแทนรูปแบบการทำงาน Manual ได้					
	1.4 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถเพิ่มประสิทธิภาพในการทำงานได้จากรูปแบบการทำงาน Manual					
	1.5 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถนำไปต่อยอดเพื่อประยุกต์ใช้ร่วมกับกลยุทธ์การบริหารจัดการทางเทคโนโลยีสารสนเทศได้					
	1.6 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถทำงานได้อย่างมีประสิทธิภาพ ในขณะที่ใช้จำนวนพนักงานปฏิบัติการน้อยลง					

ลำดับ	รายการสอบถาม	ระดับคะแนน				
		5 – มากที่สุด	4 – มาก	3 – ปานกลาง	2 – น้อย	1 – น้อยที่สุด
2	**การลดระยะเวลาและขั้นตอนการทำงานจากการใช้งาน**	-	-	-	-	-
	2.1 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถช่วยลดระยะเวลาในขั้นตอนการทำงานเพื่อนำไปใช้งานจริง					
	2.2 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถช่วยลดขั้นตอนการทำงานเพื่อนำไปใช้งานจริง					
	2.3 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถช่วยลดระยะเวลาในทุกขั้นตอนที่นำมาใช้ทดแทนการทำงานในรูปแบบ Manual					
	2.4 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ช่วยเพิ่มความสามารถของผลผลิตซอฟต์แวร์ในเชิงกลยุทธ์จากการใช้ระยะเวลาที่ลดลง					
	2.5 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ช่วยเพิ่มความรวดเร็วให้แก่ผลผลิตซอฟต์แวร์ในการนำออกไปใช้งานกับผู้ใช้งาน					
3	**การทำงานตามเป้าหมายของ Service Level Agreements**	-	-	-	-	-
	3.1 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps สามารถทำงานได้ตาม					

ลำดับ	รายการสอบถาม	ระดับคะแนน				
		5 – มากที่สุด	4 – มาก	3 – ปานกลาง	2 – น้อย	1 – น้อยที่สุด
	เป้าหมายที่กำหนดไว้ใน Service Level Agreements					
	3.2 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ช่วยเพิ่มเปอร์เซ็นต์ความสำเร็จตาม Service Level Agreements ที่กำหนดไว้					
	3.3 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ช่วยเพิ่มประสิทธิภาพความสำเร็จจากการทำงานตามแผนงานของฝ่ายงาน ซึ่งมีส่วนจากเปอร์เซ็นต์ความสำเร็จตาม Service Level Agreements ที่กำหนดไว้					
	3.4 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ช่วยลดปัญหาที่เกิดขึ้นจากความไม่เป็นไปตามเป้าหมายของ Service Level Agreements ที่กำหนดไว้					
	3.5 Automation Server ภายใต้แนวคิด CI/CD ของกระบวนการ DevOps ทำให้ช่วยทำให้ภาพรวมของการปฏิบัติงานตาม Service Level Agreements ดีขึ้นอย่างเห็นได้ชัด					

ประวัติผู้วิจัย

ชื่อ	นายเมธัส คำจาด
วัน เดือน ปีเกิด	15 มิถุนายน 2531
สถานที่เกิด	โรงพยาบาลราชวิถี กรุงเทพมหานคร
ประวัติการศึกษา	- บริหารธุรกิจบัณฑิต (บธ.บ.) สาขาวิชาคอมพิวเตอร์ธุรกิจ มหาวิทยาลัยศรีปทุม กรุงเทพมหานคร - ศึกษาศาสตรมหาบัณฑิต (ศษ.ม.) สาขาวิชาการบริหารการศึกษา มหาวิทยาลัยนอร์ทกรุงเทพ กรุงเทพมหานคร
สถานที่ทำงาน	สมาคมตลาดตราสารหนี้ไทย
ตำแหน่ง	เจ้าหน้าที่ (Full Stack Developer) ฝ่ายเทคโนโลยีสารสนเทศ