**1. DevOps คืออะไร มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร**

Ebert และคณะ [1] ได้ให้ความหมายว่า “DevOps คือการพัฒนาระบบ และการจัดเตรียมระบบธุรกิจที่มีความเร็วและมีความยืดหยุ่น เพื่อให้การดำเนินการมีประสิทธิภาพ”

DevOps คือรูปแบบวิธีการปฏิบัติ วัฒนธรรม และกระบวนการต่างๆ เพื่อแก้ไขปัญหาที่เกิดจากความขัดแย้งระหว่าง Development และ Operations ดังที่กล่าวมาแล้วในหัวข้อด้านบน รวมถึงช่วยเพิ่มประสิทธิภาพในการทำงานให้สามารถผลิต software ออกสู่ตลาดได้รวดเร็วขึ้น มีคุณภาพและเสถียรภาพมากขึ้น ในขณะที่ cost ลดลง เพื่อให้บริษัทสำเร็จตามเป้าหมายที่วางไว้ [2]

ทั้งนี้ DevOps เกิดขึ้นมาเพื่อแก้ปัญหาการทำงานระหว่าง ฝ่ายงานนักพัฒนา (Development) ซึ่งจะมุ่งพัฒนาระบบที่มีการเปลี่ยนแปลงอย่างรวดเร็ว เพื่อตอบสนองต่อความต้องการของผู้ใช้ และฝ่ายเจ้าหน้าที่ส่วนกลาง (Oparations) ซึ่งเป็นฝ่ายที่ทำการ deploy โดยจะมุ่งไปที่ความเสถียร มั่นคง และให้ความสำคัญกับความปลอยภัยของลูกค้า

ซึ่งจากที่เป้าหมายของแต่ละฝ่ายมีความขัดแย้งกัน ซึ่งโดยทั่วไปบริษัทจะมุ่งเน้นไปที่การส่งมอบงานให้เร็วที่สุดเพื่อที่จะได้นำเงินมาหมุนเวียนได้ แต่อาจจะส่งผลกระทบในภายหลังโดยเกิดเป็นปัญหาเก่าสะสมจากระบบที่ไม่เสถียร โดยที่ DevOps จะถูนำมาใช้ในการแก้ไขปัญหานี้ เพื่อให้สามารถลดระยะเวลาในการผลิตซอฟต์แวร์ แต่ยังสามารถคงคุณภาพเอาไว้ได้ โดยหลักสำคัญของ

DevOps จะประกอบไปด้วย 3 ข้อ ดังนี้

1. หลักการของ flow

จะเป็นการปรับปรุงระบบการทำงานโดยจะทำให้ flow ของการทำงานไหลเวียนได้รวดเร็ว โดยการแบ่งงานออกเป็นชิ้น ย่อย เพื่อให้เพื่อให้แต่ละฝ่ายสามารถทำงานได้ง่ายขึ้น และจะทำให้สามารถทำการรวบบ่อยๆ (Continuous integration) และยังสามารถจำกัดงานในระบบ (Limit work in process) ส่งผลให้สามารถลดข้อผิดพลาด (Defact) ของงานได้ ทำให้สามารถที่จะส่งออกงาน (Deploy) ได้บ่อยขึ้น

2. หลักการของ feedback

การรับข้อผิดพลาด (Feedback) ให้เร็ว และแก้ปัญหาให้เร็ว จะทำให้งานเกิดประสิทธิภาพ โดยจะสามารถลดปัญหาได้ก็ควรที่จะมีการสร้างการทดสอบโดยอัตโนมัติ (Automate test) เพื่อแก้ไขปัญหาได้ทันที่ และสามารถติดตามข้อผิดพลาดของงานได้ตลอดเวลา

3. หลักการทดลองและการเรียนรู้อย่างต่อเนื่อง

เป็นการนำองค์ความรู้ใหม่ ๆ มาใช้ในการปรับปรุงระบบเรื่อย ๆ เพื่อให้ระบบมีประสิทธิภาพเพิ่มมากขึ้น และเมื่อเกิดข้อผิดพลาดก็จะแก้ปัญหาได้อย่างรวดเร็ว ส่งผลให้สามารถเรียนรู้ได้อย่างรวดเร็ว

**2. CI/CD คืออะไร มีความสัมพันธ์กับ DevOps อย่างไร**

CI/CD เป็นกระบวนการหรือแนวทางที่จะช่วยในการพัฒนาซอฟต์แวร์ เพื่อให้กระบวนการผลิตซอฟต์แวร์ มีประสิทธิภาพ [3] ซึ่ง CI/CD สามารถแยกออกเป็น CI และ CD โดยมีความหมายดังนี้

CI ย่อมาจาก Continuous Integration เป็นกระบวนการที่จะต้องนำงานที่ผ่านการย่อยเป็นชิ้นส่วนเล็ก ๆ จากกระบวนการของ DevOps มารวมกัน เพื่อให้แน่ใจว่างานที่ได้นั้น สามารถทำงานได้จริง และไม่มีข้อผิดพลาด โดยการทำ CI นั้นควรจะทำบ่อย ๆ เพื่อป้องกันข้อผิดพลาดซึ่งอาจเกิดขึ้นได้โดยเร็ว

CD ย่อมาจาก Continuous Delivery หรือ Continuous Deployment เป็นกระบวนการส่งมอบงาน การส่งมอบงานที่สามารถใช้งานได้จริง เพื่อให้ลูกค้าตรวจสอบ การส่งมอบงานที่บ่อย จะก่อให้เกิดความน่าเชื่อถือจากลูกค้า เนื่องจากมีตัวซอฟท์แวร์ ที่เป็นชิ้นเป็นอัน และบริษัทยังสามารถจดสัญญาได้ในลักษณะของรอบ เมื่อส่งงานแต่ละรอบสำเร็จจะทำให้ได้รับเงินตอบแทน เพื่อนำมาเป็นสภาพคล่องได้ โดย CD สามารถแยกเป็น 2 ประเภทหลัก ๆ จากพฤติกรรมการส่งมอบงาน คือ

Continuous Delivery และ Continuous Deployment

Continuous Delivery เป็นการส่งมอบงานในลักษณะของ manual ซึ่งการส่งมอบจะต้องผ่านการอนุมัติจากผู้เกี่ยวข้องก่อน เมื่อผ่านแล้ว จึงจะสามารถดำเนินการ Deploy ขึ้น Server ได้

Continuous Deployment เป็นการส่งมอบงานในลักษณะของ Automate เพื่อให้การส่งงานนั้น สามารถทำได้รวดเร็ว โดยสามารถ Deploy ได้เลยโดยไม่ต้องผ่านการอนุมัติใด ๆ

CI/CD มีความสัมพันธ์กับ DevOps ในเรื่องของการพัฒนาซอฟท์แวร์ จากหลักการที่ DevOps จะใช้ในการแก้ไขปัญหาประสิทธิภาพของงานและความเร็วในการผลิต ซึ่งทั้งสองนั้นมักจะแปรผกผันกัน กล่าวคือ หากต้องการระบบที่มีประสิทธิภาพสูง จะต้องแลกกับระยะเวลาในการผลิตที่ยาวนาน เพราะเหตุนี้ DevOps จึงเป็นกระบวนการที่ช่วยให้สามารถทำทั้งสองนี้ได้ โดยการแยกงานออกเป็น ชิ้น ๆ ซึ่งเมื่อแยกแล้ว การนำ CI/CD มาช่วยในการรวมกลับงานบ่อย ๆ การส่งมอบงานบ่อย ๆ เพื่อให้งานนั้นมีประสิทธิภาพ และสามารถส่งมอบงานได้เร็ว จากข้อผิดพลาดที่น้อยลงอันเนื่องมาจากกระบวนการของ CI/CD

**3. หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์จะต้องทำอย่างไรบ้าง**

การนำแนวคิดของ Devops มาใช้ในการพัฒนาซอฟต์แวร์นั้น องค์กรต้องทำการทดลองและประยุกต์ใช้ให้เหมาะกับองค์กรตัวเอง [4] โดยจะมีกระบวนการดังนี้

1 เชิญชวนให้ฝ่าย Oparation เข้ามาร่วมทำงานกับ Delvelopment เพื่อให้ทีมทั้งสองเกิดปฏิสัมพันธ์ที่มาขึ้น

2 การวิเคราะห์งานร่วมกัน ซึ่งในกระบวนการของ DevOps จะเป็นการทำงานร่วมกันระหว่างฝ่าย Oparation และ Delvelopment เมื่อทำงานร่วมกันได้ก็จะทำให้งานมีความลื่นไหล

3 สร้างกระบวนการทดสอบแบบอัตโนมัติ ตั้งแต่การดึงข้อมูลจาก Repository กลางและทำการรันทดสอบโดยอัตโนมัติ เพื่อให้งานเกิดความรวดเร็วมากขึ้น

4 สร้างแผนในการ Deploy ซึ่งจะทำสามารถคาดการณ์ข้อผิดพลาดที่จะเกิดขึ้น โดยการวางแผนไม่ให้ระหว่างการทำงานจะไม่เกิดข้อผิดพลาด

5 การคาดการณ์ปัญหา ระหว่างการทำงานนั้นอาจจะเกิดปัญหาขึ้นได้ทุกเมื่อ การหาข้อผิดพลาดที่เกิดบ่อย ๆ จะทำให้การทำงานมีประสิทธิภาพเพิ่มมากขึ้น

6 การระบุปัญหาที่น่าจะทำให้งานเกิดความล่าช้า ซึ่งเนื่องมาจาก ระบบบางอย่างต้องใช้เวลานานในการประมวนผล และการตั้งค่า เพื่อไม่ให้งานส่วนอื่นกระทบ ควรแยกงานที่สำคัญเป็นอันดับแรก

**4. อ้างอิง**

[1] Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). **DevOps**. *Ieee Software*, *33*(3), 94-100.

[2] ปริวัตร ศักดิ์นิมิตวงค์. (2017). **Learn DevOps ตอนที่ 2 : DevOps คืออะไร ?**. สืบค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก /////https://medium.com/@pariwat\_s/ learn-devops-ตอนที่-2-devops-คืออะไร-18ac48d73625#0e8b

[3] ศรีนิกร. (2020). **ทำความรู้จักกับ CI/CD services บน Google Cloud**. สืบค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก /////https://cloud-ace.co.th/blog/w0w7e1-ci-cd-services-google-cloud

[4] สมเกียรติ ภูสูงเนิน. (2015). **7 วิธีสำหรับนำ DevOps เข้ามาใช้ในองค์กร**. สืบค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก /////https://www.somkiat.cc/start-with-devops/