**(1) DevOps คืออะไร มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร**

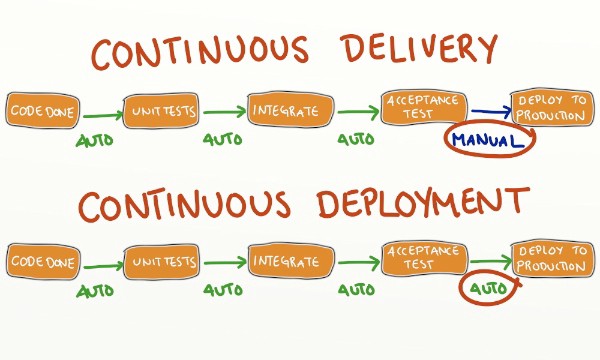
DevOps คือ แนวคิด แนวปฏิบัติ วัฒนธรรม และกระบวนการต่างๆที่ช่วยแก้ไขปัญหาความขัดแย้งระหว่างทีมพัฒนา (Development) และทีมปฏิบัติ (Operations) [1] เพื่อให้เอางานขึ้น Production ให้เร็วที่สุด รวมถึงส่งมอบงานที่มีคุณภาพให้ได้อย่างต่อเนื่องเป็นระบบ [2]

DevOps ช่วยเพิ่มประสิทธิภาพขององค์กรให้สามารถผลิตซอฟต์แวร์ออกสู่ตลาดได้อย่างน่าเชื่อถือในความเร็วที่มากขึ้น สามารถปรับตัวต่อการเปลี่ยนแปลงของตลาดได้ดี การสื่อสารและการทำงานร่วมกันที่เพิ่มขึ้นในองค์กร ซึ่งทั้งนี้ องค์กรอาจต้องมีการปรับปรุงขั้นตอนและเปลี่ยนกระบวนการพัฒนาซอฟต์แวร์และจัดการโครงสร้างพื้นฐานให้ทำงานอัตโนมัติโดยนำลำดับการทำงานและความรับผิดชอบของทีมพัฒนาและทีมปฏิบัติการเข้ามาอยู่ด้วยกันและรับ Feedback กลับไปเพื่อปรับปรุงให้ดีขึ้นเรื่อย ๆ ทีมเหล่านี้ก็จะกำหนดบรรทัดฐานเชิงวัฒนธรรมที่เข้มแข็งเกี่ยวกับการแบ่งปันข้อมูลและการอำนวยความสะดวกในการสื่อสารทำให้ทุกส่วนขององค์กรปรับตัวสอดคล้องกับเป้าหมายและโครงการได้มากขึ้น [3]

**(2) CI/CD คืออะไร มีความสัมพันธ์กับ DevOps อย่างไร**

Continuous Integration (CI) คือ กระบวนการที่ใช้สำหรับรวบรวมซอฟต์แวร์ที่มีการพัฒนาแยกส่วนกันของคนในทีมพัฒนานำมารวมเข้าด้วยกัน เพื่อเป็นการทำให้มั่นใจ และป้องกันข้อผิดพลาดที่อาจะเกิดขึ้นได้ในอนาคต เนื่องจากเป็นการพัฒนาจากโปรแกรมเมอร์หลายคนจึงต้องมีการทดสอบความเข้ากันได้ของงานแต่ละส่วนโดยอัตโนมัติ โดยการเขียน test script [4]

Continuous Delivery และ Continuous Deployment (CD) ทั้งสองคำนี้มีขั้นตอนการพัฒนาที่แตกต่างกันดังภาพที่ 2 โดย Continuous Deployment คือ ทุกขั้นตอนตั้งแต่ Compile build ไปจนถึง Deploy ขึ้น Production แบบอัตโนมัติทั้งหมด Continuous Delivery มีการทำทุกขั้นตอนคล้ายกันกับ Continuous Deployment แต่ไม่มีการ Deploy ขึ้น Production ในทันทีจะต้องได้รับการอนุมัติหรือการตัดสินใจกันก่อนจากทาง Business ซึ่งเป็นการทำงานแบบ Manual หรือจะเป็นแบบ One Click Deploy ก็ได้



**ภาพที่ 2** ขั้นตอนการพัฒนาระบบของ Continuous Delivery และ Continuous Delivery

(ที่มา: https://blog.appsynth.net/continuous-delivery-of-valuable-software-9855bff30cb5)

กระบวนการพัฒนาซอฟต์แวร์โดยใช้ CI/CD เชื่อมช่องว่างระหว่างทีมพัฒนาและทีมปฏิบัติเขาด้วยกัน โดยทำให้การสร้างซอฟต์แวร์ การทดสอบซอฟต์แวร์ และการวางระบบซอฟต์แวร์เป็นไปได้โดยอัตโนมัติ CI/CD เป็นโมเดลที่ครอบคลุมการทำงานเพื่อเพิ่มความเร็วในการส่งมอบซอฟต์แวร์ออกสู่ตลาด ซึ่งนับเป็นแกนหลักสำคัญของกระบวนการ DevOps ที่ทันสมัย นอกจากนี้ การมีระบบอัตโนมัติยังเป็นตัวช่วยที่ทำให้ทีมพัฒนาสามารถปรับ เพิ่ม หรือเปลี่ยนแปลงฟีเจอร์ต่างๆ บนซอฟต์แวร์ได้ตามความต้องการ ในขณะที่ทีมปฏิบัติยังคงสามารถปฏิบัติงานได้อย่างมีเสถียรภาพ [5]

**(3) หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์จะต้องทำอย่างไรบ้าง**

บริษัทสามารถนำซอฟต์แวร์ต่าง ๆ มาใช้ในกระบวนการทำงานหรือที่เราเรียกว่า DevOps เริ่มตั้งแต่ลักษณะวิธีการทำงาน อาจเปลี่ยนมาใช้ Agile เพื่อให้เหมาะสมกับบริบทของการทำงาน ต่อมาเรื่องการสื่อสาร เนื่องจากช่วงนี้มีสถานการณ์ โควิด-19 อาจมีความจำเป็นต้องทำงานจากที่บ้านเป็นส่วนใหญ่ทำให้ต้องมีซอฟต์แวร์ติดต่อสื่อสารกันระยะไกล ยกตัวอย่างเช่น Google Meet หรือ Zoom จากนั้นระหว่างวันก็ควรมีการพูดคุยกัน อาจใช้เครื่องมืออย่าง Slack หรือ Microsoft Team ในการสื่อสารกัน และวางแผนการทำงานกันด้วยซอฟต์แวร์อย่าง Jira หรือ Asana เป็นต้น ช่วงของการเขียนโค้ด ทีมก็สามารถนำซอฟต์แวร์อย่าง Git มาเพื่อจัดการกับโค้ดที่กำลังทำอยู่ เพื่อลดปัญหาความไม่เข้ากันของและความสดใหม่ของโค้ดได้เป็นอย่างดี ต่อมาเมื่อพัฒนาโปรแกรมแล้วก็สามารถนำโค้ดที่อยู่ใน Git มา Compile โปรแกรม เทสโปรแกรม และนำขึ้นสู่ระบบจริงด้วยระบบ CI/CD อย่าง Jenkins หรือ GitHub Actions การใช้ CI/CD จะทำงานแบบอัตโนมัติซึ่งจะทำให้ทีมพัฒนาสะดวกขึ้นเป็นอย่างมากในการ Deploy ระบบ ในการ run ระบบก็สามารถใช้ Container อย่างเช่น Docker Swarm หรือ Kubernetes เข้ามาแทนที่ระบบเดิมอย่าง VM ได้ ลดปัญหาความไม่เข้ากันของซอฟต์แวร์ที่อยู่ใน OS ของ Host และมีความสามารถในการทำ Scalability หรือการเพิ่มลดจำนวนของ Instance เพื่อให้พอดีกับความต้องการของผู้ใช้ในแต่ละช่วงเวลาได้อย่างเหมาะสมโดยอัตโนมัติ ต่างจากระบบเดิมที่จะต้องสร้าง VM เพิ่มเองและตั้งกฎของ Load balancer ใหม่ ในขณะที่โปรแกรมกำลังใช้งานในระบบจริงก็อาจมีปัญหาเกิดขึ้น โปรแกรมเมอร์จำเป็นต้องรู้ถึงที่มาของปัญหานั้น ๆ ว่าเกิดขึ้นที่ใดและเกิดขึ้นได้อย่างไร จึงมีซอฟต์แวร์สำหรับบันทึกความผิดพลาดของการทำงาน บันทึก Traffic ต่าง ๆ ในโปรแกรมและวิเคราะห์ที่มาของปัญหาได้ทันที อย่างเช่น Sentry และ Grafana เป็นต้น

**อ้างอิง**

[1] Saknimitwong, P. (2017). **Learn DevOps ตอนที่ 2 : DevOps คืออะไร ?.** ค้นเมื่อ26 มกราคม 2564, จากhttps://medium.com/@pariwat\_s/learn-devops-ตอนที่-2-devops-คืออะไร-18ac48d73625

[2] สารานุกรมเสรีวิกิิพีเดีย. (มปป). **DevOps.** ค้นเมื่อ26 มกราคม 2564, จากhttps://en.wikipedia.org/wiki/DevOps

[3] บริการประมวลผลบนระบบคลาวด์. (มปป). **DevOps คืออะไร.** ค้นเมื่อ26 มกราคม 2564, จาก https://aws.amazon.com/th/devops/what-is-devops/

[4] บริษัท ซอฟท์เมลท์ จำกัด. (มปป). **การพัฒนาระบบตามแนวทาง CI/CD และ DevOps คืออะไร?.** ค้นเมื่อ27 มกราคม 2564, จากhttps://www.softmelt.com/article.php?id=664

[5] ทีมงาน TechTalkThai. (2562). **Sponsored Webinar: Automate DevOps & SecOps deployments with CI/CD Pipeline Integration.** ค้นเมื่อ27 มกราคม 2564, จากhttps://www.techtalkthai.com/sponsored-webinar-automate-devops-secops-deployments-with-ci-cd-pipeline-integration/