**DevOps คืออะไร มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร**

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

**CI/CD คืออะไร มีความสัมพันธ์กับ DevOps อย่างไร**

Continuous Integration (CI) คือ กระบวนท่าที่ใช้สำหรับการรวบรวมซอฟแวร์ที่มีการพัฒนาแยกส่วนกันอย่างอัตโนมัติ อาจจะโดยหนึ่งหรือหลายนักพัฒนาก็ตามที สุดท้ายแล้วซอฟแวร์ที่พัฒนาชิ้นเล็กๆ ที่พัฒนาขึ้นมาจะต้องนำมารวมกันเป็นชิ้นใหญ่หนึ่งชิ้น จะทำอย่างไรให้มั่นใจได้ว่า ไม่มีชิ้นส่วนใดที่จะส่งผลให้ชิ้นส่วนอื่นๆ พังเสียหาย เนื่องจากเป็นการพัฒนาโดยโปรแกรมเมอร์หลายคน

ซึ่งเป็นไปได้ว่าจะมี bug หลุดมาจากส่วนใดส่วนหนึ่ง แล้วเราจะป้องกันได้อย่างไรละ ดังนั้นจึงต้องมีการเขียน script test ที่คอยทดสอบความเข้ากันได้ของแต่ละชิ้นส่วนโดยอัตโนมัตินั่นเอง โดยการ Testing จะเริ่มตั้งแต่ Unit Testing ซึ่งสร้างจากทีมพัฒนา และเป็นส่วนจะใช้ตรวจสอบว่าสิ่งที่ทีมพัฒนายังทำงานถูกต้องและจะใช้เวลาช่วงสั้น ๆ เท่านั้น

DevOps คือ การรวมคำว่า Developer กับ Operator เข้าด้วยกัน โดยสมัยก่อน Developer เป็นคนพัฒนา code แต่ไม่มีสิทธิ deploy code ทำให้การทำงานยุ่งยาก เกิดความผิดพลาด และใช้เวลานานในการ deploy แต่ละครั้ง

เพื่อแก้ปัญหาในการ deploy code จึงเกิดเป็นแนวทาง CI/CD โดยทำงานตั้งแต่การ Plan, Code, Build, Test, Release, Deploy, Operate, Monitor หรือบางทีเรียกสั้นๆ ว่า Pipeline

**หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์จะต้องทำอย่างไรบ้าง**

**ขั้นตอนที่ 1** ระบุปัญหาก่อนว่ามีอะไรบ้าง

จากนั้นจึงทำการเรียกลำดับตามความสำคัญหรือผลกระทบที่เกิดขึ้นว่าจะควรแก้ไขอะไรก่อนหลัง

**ขั้นตอนที่ 2** ลดความหลากหลาย

สิ่งที่มักพบเจอบ่อยๆ ในการพัฒนา software คือ ความหลากหลายของการทำงาน ไม่ว่าจะเป็น

-ขั้นตอนการทำงาน

-Environment ต่าง ๆ ที่ใช้งาน

-เครื่องมือที่หลายหลาย ตามใจใครหลาย ๆ คน

-การ configuration ที่ไร้ทิศทาง

-ข้อมูลที่หลายหลายชนิด อยู่หลายที่

**ขั้นตอนที่ 3** ต้องรู้ขั้นตอนการทำงานตั้งแต่ต้นจนจบ

เราไม่สามารถจะปรับปรุงอะไรได้เลย หากเราไม่รู้กระบวนการทำงานตั้งแต่ต้นจนจบ

เนื่องจากเรามักจะมีคนที่รู้เฉพาะจุดหรือบางส่วน ดังนั้นสิ่งที่ควรทำคือ อะไรที่ไม่ทราบ ก็ควรศึกษาจนกว่าจะทราบในเรื่องนั้นๆ จากนั้นทำการสร้างเอกสาร เพื่ออธิบายการทำงานตั้งแต่ต้นจนจบไว้ จากนั้น share ให้ทุกคนที่เกี่ยวข้อง จะได้ทำความเข้าใจแก้ไขหรือปรับปรุงต่อไป ซึ่งทั้งหมดนี้จะทำให้เราเห็นว่า การทำงานปัจจุบันเป็นอย่างไร ตรงไหนมีปัญหาบ้าง

**ขั้นตอนที่ 4** เข้าสู่กระบวนการพัฒนาและส่งมอบ software

การพัฒนาและส่งมอบ software ที่ดีประกอบไปด้วย

-การจัดการ source code หรือ Source Control Management (SCM)

-กระบวนการ build ของ software

-กระบวนการ deploy software

-การ configuration หรือ provisioning environment ต่าง ๆ สำหรับการ deploy software

ซึ่งเราอาจจะต้องกำหนดการทำงานเหล่านี้ให้ชัดเจนหรือดีที่สุดคือ ช่วยกันสร้างระบบที่ทำงานแบบอัตโนมัติขึ้นมาเพื่อลดงานที่ต้องมีคนเข้าไปยุ่งเกี่ยวให้มากที่สุด เพราะว่ามันคืองานที่ทำซ้ำ ๆ บ่อยมากและที่สำคัญคือ เพื่อลดปัญหาที่เกิดจากคนให้มากที่สุด

**ขั้นตอนที่ 5** กระบวนการทดสอบแบบอัตโนมัติ

การทดสอบนั้นเป็นหัวใจของการพัฒนา software และ DevOps เนื่องจากมันสะท้อนในเรื่องของคุณภาพ software ที่ส่งมอบมากพอสมควรโดยการทดสอบควรต้องเป็นแบบอัตโนมัติให้ได้มากที่สุดเพื่อช่วย validate ว่า software ของเรายังคงทำงานได้อย่างถูกต้องตามที่คาดหวังที่สำคัญต้องทำงานได้รวดเร็วด้วย มิเช่นนั้น ปัญหาต่าง ๆ ก็ยังคงอยู่เสมอ