

DevOps

กระบวนการผลิตซอฟต์แวร์นั้น มักจะมีการขัดแย้งกันระหว่าง 2 ฝ่ายหลัก ๆ นั่นคือ ฝ่าย Development กับ ฝ่าย Operations ที่แบ่งตามหน้าที่หลัก ๆ ในการผลิตซอฟต์แวร์นั้นคือ ฝ่าย Dev ต้องผลิตซอฟต์แวร์อย่างรวดเร็วส่งมอบให้ทันเวลา และฝ่าย Ops ต้องดูแลด้านความเสถียรของซอฟต์แวร์เพื่อมีความผิดพลาดน้อยที่สุด และในกระบวนการผลิตก็จะต้องเลือกระหว่างความเสถียรภาพ หรือความเร็ว [1]

การเลือกความเสถียรภาพ เพื่อให้งานที่ออกมามีจุดบกพร่องให้น้อยที่สุด เมื่อส่งมอบแล้วได้งานที่มีประสิทธิภาพมีความเสถียรและปลอดภัย เมื่อส่งมอบแล้วเจอปัญหาที่น้อย ส่งมอบเดียวจบเสียเวลามาตามแก้ปัญหาในภายหลังที่น้อยลง แต่มีปัญหาที่กระบวนการผลิตนั้น จะเป็นไปอย่างเชื่องช้า กว่าจะได้ส่งมอบฟีเจอร์อะไรซักหนึ่งอย่าง

การเลือกความเร็ว เพื่อให้ส่งมอบได้เร็ว ๆ สามารถผลิตฟีเจอร์ใหม่ ๆ ได้มากขึ้น แต่ต้องแลกกับความเสถียร การส่งมอบรอบนี้อาจมีจุดผิดพลาดเล็ก ๆ น้อย ๆ ที่มองไม่เห็น แต่เมื่อเวลาผ่านไปจุดผิดพลาดทับถมมากขึ้นเรื่อย ๆ จนเป็นปัญหาใหญ่อาจทำให้เสียเวลาในการแก้ปัญหามากขึ้นในภายหลัง การเปลี่ยนแปลงเพื่อเพิ่มฟีเจอร์ก็จะเป็นไปได้ยากขึ้นด้วย

DevOps เป็นแนวคิดการทำงานที่เปลี่ยนรูปแบบระบบการทำงานของทีมโดยมีการเปลี่ยนแปลงที่สำคัญคือ เปลี่ยนเป้าหมายการทำงานจากรับผิดชอบงานของตนเองเป็นรับผิดชอบทั้งระบบ จากเดิมที่แต่ละฝ่ายมีเป้าหมายการทำงานไม่เหมือนกันเปลี่ยนให้กลายเป็นทีมเดียวกันทำงานร่วมกันมากขึ้น มีการรับผิดชอบร่วมกันมากขึ้น ทำให้ทั้ง 2 ฝ่ายสื่อสารกันเข้าใจปัญหาของแต่ละฝ่าย ทุกฝ่ายเข้าใจภาพรวมของงานมากขึ้นทำให้ปัญหา และหน้าที่การแก้ปัญหาไม่ตกไปอยู่ฝ่ายใดฝ่ายหนึ่งเพียงฝ่ายเดียว เช่นฝ่าย Dev เข้าใจว่าสิ่งที่ผู้ใช้ต้องการจริง ๆ คืออะไร ไม่ใช่ทำตามเพียงคำสั่งที่ได้รับเข้ามา ฝ่าย Ops เข้าใจการทำงานของโปรแกรมมากขึ้นสามารถรู้ได้ทันทีว่าปัญหาของผู้ใช้เกิดขึ้นที่ตรงไหน

ดังนั้น DevOps จึงมาจากการรวมคำว่า Developer กับ Operator เข้าด้วยกัน [2] จากที่ Developer เป็นคนพัฒนา code แต่ไม่มีสิทธิ deploy code ทำให้การทำงานยุ่งยาก เกิดความผิดพลาด และใช้เวลานานในการ deploy แต่ละครั้ง และเพื่อแก้ปัญหาในการ deploy code จึงเกิดเป็นแนวทาง CI/CD โดยทำงานตั้งแต่การ Plan, Code, Build, Test, Release, Deploy, Operate, Monitor หรือบางที่เรียกสั้นๆ ว่า Pipeline

CI/CD

CI/CD [3] คือแนวทางในการทำงานของ DevOps

Continuous Integration(CI) คือ กระบวนการรวม source code ของคนในทีมพัฒนาเข้าด้วยกัน และมีการ test ด้วย test script เพื่อให้แน่ใจว่าไม่มี error ในส่วนใดๆ ของโปรแกรม แล้วถึงทำการ commit ไปที่ branch master อีกต่อหนึ่ง

Continuous Delivery และ Continuous Deployment (CD)

Continuous Deployment จะทำทุกขั้นตอน ตั้งแต่ complie build ไปจนถึง deploy ขึ้น production แบบอัตโนมัติทั้งหมด

และในส่วนของ Continuous Delivery มีการทำทุกขั้นตอนคล้ายกันกับ Continuous Deployment ต่างกันตรงที่จะไม่มีการ deploy ขึ้น production ขึ้นในทันที แต่จะเป็นการทำ manual deploy หรือจะเป็นแบบ one click deploy ก็ได้ หลังจาก QA หรือ ฝ่าย Business พอใจในตัว product ที่ทีมทำออกมา

หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์

ต้องทำการคุย วางแผนระบบงาน เครื่องมือต่าง ๆ ที่ใช้ให้ทุกคนเข้าใจตรงกัน แล้วทำการสร้าง project หนึ่งสำหรับทดสอบขึ้นมาเพื่อให้ทุกคนได้เข้าใจถึงระบบและใช้เครื่องมือต่าง ๆ ได้อย่างไม่มีปัญหา

เอกสารอ้างอิง

[1] Developer Team BorntoDev Co., Ltd. (2020). **DevOps คืออียังวะ ?**. ค้นเมื่อ 2 กุมภาพันธ์ 2564, จาก

<https://www.borntodev.com/2020/05/15/devops-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B4%E0%B8%AB%E0%B8%A2%E0%B8%B1%E0%B8%87%E0%B8%A7%E0%B8%B0/>

[2] Softmelt. [ม.ป.ป.]. **การพัฒนาระบบตามแนวทาง CI/CD และ DevOps คืออะไร?**. ค้นเมื่อ 2 กุมภาพันธ์ 2564, จาก

<https://www.softmelt.com/article.php?id=664>

[3] Thraithep Junthep. (2018). **เมื่อหัดทำ CI/CD ครั้งแรก**. ค้นเมื่อ 2 กุมภาพันธ์ 2564, จาก

<https://medium.com/@thraithep.a/%E0%B9%80%E0%B8%A1%E0%B8%B7%E0%B9%88%E0%B8%AD%E0%B8%AB%E0%B8%B1%E0%B8%94%E0%B8%97%E0%B8%B3-ci-cd-%E0%B8%84%E0%B8%A3%E0%B8%B1%E0%B9%89%E0%B8%87%E0%B9%81%E0%B8%A3%E0%B8%81-f4f79151311c>