

DevOps และ CI/CD

1. DevOps คืออะไร มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร

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 หรือบางที่เรียกสั้นๆ ว่า Pipeline ซึ่งสมัยนี้ก็มี tools ต่างๆ ที่ทำหน้าที่พวกนี้เยอะมากทั้ง On-Premise และ On-Cloud ที่เรารู้จักกันดี ก็น่าจะเป็น Jenkins ที่เข้ามามีบทบาทมากในการทำ CI/CD [1] รวมถึงช่วยเพิ่มประสิทธิภาพในการทำงานให้สามารถผลิต software ออกสู่ตลาดได้รวดเร็วขึ้น มีคุณภาพและเสถียรภาพมากขึ้น ในขณะที่ cost ลดลง เพื่อให้บริษัทสำเร็จตามเป้าหมายที่วางไว้ [2]

2. CI/CD คืออะไร มีความสัมพันธ์กับ DevOps อย่างไร

CI/CD คืออะไร?

Continuous Integration (CI) คือ กระบวนการที่ใช้สำหรับการรวบรวมซอฟต์แวร์ที่มีการพัฒนาแยกส่วนกันอย่างอัตโนมัติ อาจจะโดยหนึ่งหรือหลายนักพัฒนาก็ตามที สุดท้ายแล้วซอฟต์แวร์ที่พัฒนาขึ้นเล็กๆ ที่พัฒนาขึ้นมาจะต้องนำมารวมกันเป็นชิ้นใหญ่หนึ่งชิ้น จะทำอย่างไรให้มั่นใจได้ว่า ไม่มีชิ้นส่วนใดที่จะส่งผลให้ชิ้นส่วนอื่นๆ พังเสียหาย เนื่องจากการพัฒนาโดยโปรแกรมเมอร์หลายคน ซึ่งเป็นไปได้ว่าจะมี bug หลุดมาจากส่วนใดส่วนหนึ่ง แล้วเราจะป้องกันได้อย่างไรละ ดังนั้นจึงต้องมีการเขียน script test ที่คอยทดสอบความเข้ากันได้ของแต่ละชิ้นส่วนโดยอัตโนมัตินั่นเอง โดยการ Testing จะเริ่มตั้งแต่ Unit Testing ซึ่งสร้างจากทีมพัฒนา และเป็นส่วนจะใช้ตรวจสอบว่าสิ่งที่ทีมพัฒนายังทำงานถูกต้องและจะใช้เวลาช่วงสั้น ๆ เท่านั้น โดยในโลกของการพัฒนานั้น มักใช้ Build Server มาช่วยเพื่อให้เป้าหมายที่ตั้งไว้สำเร็จ กล่าวคือ จะเริ่มทำการ Integration กันตั้งแต่เมื่อมีการเปลี่ยนแปลง Source Code ที่ Repository กลาง ระบบจะทำการตรวจสอบ Code หลังจากการเปลี่ยนแปลงว่าทำงานร่วมกันได้หรือไม่ตั้งแต่ Compile, Testing [3]

Continuous Delivery และ Continuous Deployment (CD) คืออะไร

Continuous Deployment คือ ในทุกๆ ขั้นตอนจนถึงการ deployment ขึ้น production จะทำแบบอัตโนมัติทั้งหมด ส่วน Continuous Delivery คือ การงานต่างๆ ใน deployment pipeline นั้น จะเริ่มต้นทำงานตั้งแต่การ compile, build ไปจนถึงขั้นตอนการทดสอบต่างๆ เช่น Acceptance test เป็นแบบอัตโนมัติทั้งหมด ส่วนในขั้นตอนการ deployment ขึ้น production นั้น จะต้องได้รับการอนุมัติหรือการตัดสินใจกันก่อนจากทาง Business ซึ่งเป็นการทำงานแบบ manual นั่นเอง หรืออาจจะเป็น One Click Deploy ก็ได้ [3]

3. หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์จะต้องทำอย่างไรบ้าง

ขั้นตอนที่ 0 ระบุปัญหาก่อนว่ามีอะไรบ้าง จากนั้นจึงทำการเรียกลำดับตามความสำคัญหรือผลกระทบที่เกิดขึ้นว่าจะแก้ไขอะไรก่อนหลัง ค่อย ๆ แก้ไขทีละเรื่อง

ขั้นตอนที่ 1 ลดความหลากหลาย สิ่งที่พบเจอบ่อยมาก ๆ ในการพัฒนา software คือ ความหลากหลายของการทำงาน ดังนั้นต้องลดความหลากหลาย หรือลองหาสักแนวทางสำหรับการเริ่มต้น ยังมีความหลากหลาย ยิ่งก่อให้เกิดปัญหา ดังนั้นลองดูว่าอะไรบ้างที่ตัดออกไปแล้วมันทำให้เราทำงานง่ายขึ้น

ขั้นตอนที่ 2 ต้องรู้ขั้นตอนการทำงานตั้งแต่ต้นจนจบ สิ่งที่เราควรทำคืออะไรที่ไม่รู้ทำให้รู้ จากนั้นทำการสร้างเอกสาร เพื่ออธิบายการทำงานตั้งแต่ต้นจนจบไว้ จากนั้น share ให้ทุกคนที่เกี่ยวข้อง จะได้ทำความเข้าใจ แก้ไขหรือปรับปรุงกันไป ซึ่งจะช่วยให้เราเห็นว่าการทำงานปัจจุบันเป็นอย่างไร ตรงไหนมีปัญหา

ขั้นตอนที่ 3 เข้าสู่กระบวนการพัฒนาและส่งมอบ software ต้องกำหนดการทำงานเหล่านี้ให้ชัดเจนหรือดีที่สุดคือ ช่วยกันสร้างระบบที่ทำงานแบบอัตโนมัติขึ้นมาเพื่อลดงานที่ต้องมีคนเข้าไปยุ่งเกี่ยวให้มากที่สุด เพราะว่า มันคืองานที่ทำซ้ำ ๆ บ่อยมากและที่สำคัญคือ เพื่อลดปัญหาที่เกิดจากคนให้มากที่สุด

ขั้นตอนที่ 4 กระบวนการทดสอบแบบอัตโนมัติ การทดสอบควรต้องเป็นแบบอัตโนมัติให้ได้มากที่สุด เพื่อช่วย validate ว่า software ของเรายังคงทำงานได้อย่างถูกต้องตามที่คาดหวัง ที่สำคัญต้องทำงานได้รวดเร็ว ด้วย มีเช่นนั้น ปัญหาต่าง ๆ ก็ยังคงอยู่เสมอ มันน่ากลัวมาก ๆ ถ้าเราทำการ deploy software บ่อย ๆ แต่ผิดเยอะบ่อย ๆ มันไม่น่าจะใช้ผลที่เราต้องการ [4]

แหล่งอ้างอิง

- [1] Administrator. “การพัฒนาระบบตามแนวทาง CI/CD และ DevOps คืออะไร?”
<https://www.softmelt.com/article.php?id=664> (February 27, 2021).
- [2] narissara. 2017. “การนำ Automated Testing มารวมเข้ากับ กระบวนการ CI/CD.” *Medium*.
<https://medium.com/@narissara.tan/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%99%E0%B8%B3-automated-testing-%E0%B8%A1%E0%B8%B2%E0%B8%A3%E0%B8%A7%E0%B8%A1%E0%B9%80%E0%B8%82%E0%B9%89%E0%B8%B2%E0%B8%81%E0%B8%B1%E0%B8%9A-%E0%B8%81%E0%B8%A3%E0%B8%B0%E0%B8%9A%E0%B8%A7%E0%B8%99%E0%B8%81%E0%B8%B2%E0%B8%A3-ci-cd-58c97e31e230> (February 27, 2021).
- [3] Saknimitwong, Pariwat. 2018. “Learn DevOps ตอนที่ 2 : DevOps คืออะไร ?” *Medium*.
https://medium.com/@pariwat_s/learn-devops-%E0%B8%95%E0%B8%AD%E0%B8%99%E0%B8%97%E0%B8%B5%E0%B9%88-2-devops-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-18ac48d73625 (February 27, 2021).
- [4] “แนวทางในการนำ DevOps มาใช้ปรับปรุงการพัฒนา Software.” <https://www.somkiat.cc/think-before-start-devops/> (February 27, 2021).