(1) DevOps คืออะไร มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร

ตอบ DevOps คือการผสมผสานแนวความคิดเชิงวัฒนธรรม แนวทางปฏิบัติ และเครื่องมือต่างๆ ที่ช่วยเพิ่มความสามารถขององค์กรในการส่งมอบแอปพลิเคชันและบริการอย่างรวดเร็ว โดยพัฒนาและปรับปรุงผลิตภัณฑ์ต่างๆ ให้เร็วกว่ากระบวนการการพัฒนาซอฟต์แวร์และการจัดการโครงสร้างพื้นฐานแบบดั้งเดิม ความรวดเร็วนี้ช่วยให้องค์กรสามารถให้บริการแก่ลูกค้าของตนได้ดีขึ้น และสามารถแข่งขันในตลาดได้อย่างมีประสิทธิภาพมากขึ้นในบางโมเดลของ DevOps ทีมประกันคุณภาพและทีมรักษาความปลอดภัยอาจทำงานรวมกับทีมพัฒนาและทีมปฏิบัติการอย่างใกล้ชิดยิ่งขึ้นตลอดวงจรการทำงานของแอปพลิเคชัน ในเวลาที่การรักษาความปลอดภัยเป็นจุดสำคัญของทุกคนในทีม DevOps เราจะเรียกว่า DevSecOps ทีมต่างๆ ใช้ข้อปฏิบัติในการเปลี่ยนกระบวนการต่างๆ ที่เคยทำงานแบบแมนนวลและเชื่องช้าให้ทำงานอัตโนมัติ พวกเขาใช้ชุดเทคโนโลยีและเครื่องมือต่างๆ ที่ช่วยให้พวกเขาดำเนินการและพัฒนาแอปพลิเคชันได้อย่างรวดเร็วและเชื่อถือได้ นอกจากนั้น เครื่องมือเหล่านี้ยังช่วยให้วิศวกรเหล่านี้ทำงานได้อย่างอิสระ (เช่น การปรับใช้โค้ด หรือการจัดเตรียมโครงสร้างพื้นฐาน) ที่โดยปกติแล้วจำเป็นต้องได้รับความช่วยเหลือจากทีมอื่น พร้อมทั้งยังช่วยทีมงานทำงานได้รวดเร็วยิ่งขึ้นอีกด้วย [1]

การสร้างหรือการพัฒนา นี่ก็คือขั้นตอนของการเขียนโปรแกรมขึ้นมา DevOps เข้ามาช่วยทำให้ Developers ทุกคนทำงานบน Environment แบบเดียวกัน ไม่ว่าเขาจะพัฒนาบน OS ใดก็ตามเวลา Developer ทำงานบนคอมพิวเตอร์คนละ OS กัน มักจะมีปัญหา “ทำไมโปรแกรมรันบนเครื่องเธอไม่ได้ล่ะ บนเครื่องฉันก็รันได้อยู่นะ!” ซึ่งมันไม่ควรจะเกิดขึ้นแล้วในยุคนี้ที่เรามี Docker ที่ช่วยให้ทุกคนพัฒนาโปรแกรมอยู่บนพื้นฐาน Environment เดียวกันเป๊ะ ตั้งแต่ขั้นตอนการพัฒนาไปจนถึงการส่งโปรแกรมขึ้น Production [2]

(2) CI/CD คืออะไร มีความสัมพันธ์กับ DevOps อย่างไร

ตอบ CI/CD (Continuous Integration, Continuous Delivery) เป็นกระบวนการในการทำงาน ตั้งแต่การ Plan -> Code -> Build -> Test -> Release -> Deploy -> Operate -> Monitor หรือบางทีเรียกสั้นๆ ว่า Pipeline ซึ่งสมัยนี้ ก็มี tools ต่างๆ ที่ทำหน้าที่พวกนี้ เยอะมากทั้ง On-Premise และ On-Cloud

ที่เรารู้จักกันดี ก็น่าจะเป็น Jenkins ที่เข้ามามีบทบาทมาก ในการทำ CICD ขอบเขตของการทำงาน ของ DevOps แต่ละที่เท่าที่ผมได้เคยลงไปสัมผัส จะไม่เหมือนกัน ขึ้นอยู่ว่า scope ที่ทำได้ มีระดับไหน บางที่ ก็คือทำตั้งแต่ต้นน้ำ ยันปลายน้ำ คือตั้งแต่วางแผน สร้าง Infrastructure เอง ทำ ENV ให้ Dev ใช้ เขียน Pipeline ตลอดจน ทำ Load Test, Performance Test, Security Test และระบบ Monitor & Alert เองทั้งหมด แบบนี้ก็ดีตรงที่จะรู้และเข้าใจ ในแต่ละส่วนอย่างดี ทำให้งานออกมามีประสิทธิภาพ ควบคุมได้ แต่ถ้ามี หลายๆ Project ก็คงทำแบบนี้ไม่ไหว บางที่ DevOps จะมีหน้าที่แค่ทำระบบให้ Dev มาใช้งาน แต่จะไม่มีสิทธิ ในการทำอย่างอื่นเอง แบบด้านบน แบบนี้ ก็จะทำให้ ควบคุมอะไรไม่ได้ทั้งหมด แต่ถ้ามีหลายๆ Project ก็จะรองรับการทำงาน ได้เต็มที่สำหรับแนวคิด และ Tools ในการทำงานแบบ DevOps (CI/CD) ไม่มีแบบไหนผิด แบบไหนถูก ขึ้นอยู่กับการเอามาประยุกต์ใช้งาน ให้เหมาะสมกับงานของเรา องค์กรของเรา เพื่อทำให้งาน เกิดประสิทธิภาพสูงสุด ตอบโจทย์ผู้ใช้งาน product ของเราให้ดีที่สุด .. และที่สำคัญ ต้อง Monitor ได้ ต้องมี Dashboard เอาไว้ Tracking Metric ต่างๆ

สำหรับแนวคิด และ Tools ในการทำงานแบบ DevOps (CI/CD) ไม่มีแบบไหนผิด แบบไหนถูก ขึ้นอยู่กับการเอามาประยุกต์ใช้งาน ให้เหมาะสมกับงานของเรา องค์กรของเรา เพื่อทำให้งาน เกิดประสิทธิภาพสูงสุด ตอบโจทย์ผู้ใช้งาน product ของเราให้ดีที่สุด .. และที่สำคัญ ต้อง Monitor ได้ ต้องมี Dashboard เอาไว้ Tracking Metric ต่างๆ ได้ [3]

(3) หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์จะต้องทำอย่างไรบ้าง

ขั้นตอนที่ 0 ระบุปัญหาก่อนว่ามีอะไรบ้าง ?

ตอบ จากนั้นจึงทำการเรียกลำดับตามความสำคัญหรือผลกระทบที่เกิดขึ้นว่าจะแก้ไขอะไรก่อนหลังแนะนำให้ค่อย ๆ แก้ไขทีละเรื่องถ้ามันสำคัญหมด คงจะมั่วกันน่าดู เนื่องจากเราจะไร้ทิศทางมาก ๆ ถ้าไม่มีปัญหา จะปรับปรุงไปทำไม จริงไหม ?

ขั้นตอนที่ 1 ลดความหลากหลาย

สิ่งที่พบเจอบ่อยมาก ๆ ในการพัฒนา software คือความหลากหลายของการทำงาน ไม่ว่าจะเป็นขั้นตอนการทำงานEnvironment ต่าง ๆ ที่ใช้งานเครื่องมือที่หลายหลาย ตามใจใครหลาย ๆ คนการ configuration ที่ไร้ทิศทาง

ข้อมูลที่หลายหลายชนิด อยู่หลายที่คำถามคือ ถ้ายังมีความหลากหลายแบบนี้แล้วเราจะจัดการอย่างไร ?

ยิ่งเริ่มเอาระบบการทำงานแบบอัตโนมัติเข้ามาช่วยปรับปรุง มันยิ่งลำบากหรือไม่ ?ดังนั้น ควรต้องลดความหลากหลายหรือลองหาสักแนวทางสำหรับการเริ่มต้นดีไหม ?ยิ่งมีความหลากหลาย ยิ่งก่อให้เกิดปัญหาหรือไม่ ?ดังนั้นลองดูว่า อะไรบ้างที่ตัดออกไปแล้วมันทำให้เราทำงานง่ายขึ้นบ้าง ชีวิตน่าจะดีขึ้นนะ

ขั้นตอนที่ 2 ต้องรู้ขั้นตอนการทำงานตั้งแต่ต้นจนจบ

เราไม่สามารถจะปรับปรุงอะไรได้ถ้าเราไม่รู้กระบวนการทำงานตั้งแต่ต้นจนจบเนื่องจากเรามักจะมีคนที่รู้เฉพาะจุดหรือบางส่วนคำถามหรือแล้วภาพใหญ่ละบอกเลยว่า ไม่มีใครรู้หรือรู้ก็แบบลาง ๆต้องทำการนัดประชุมกันใหญ่โต หรือใช้เวลานานมาก ๆดังนั้นสิ่งที่ควรทำคืออะไรที่ไม่รู้ทำให้รู้ (มักจะไม่รู้ว่า ไม่รู้อะไร !!)จากนั้นทำการสร้างเอกสาร เพื่ออธิบายการทำงานตั้งแต่ต้นจนจบไว้จากนั้น share ให้ทุกคนที่เกี่ยวข้องจะได้ทำความเข้าใจ แก้ไขหรือปรับปรุงกันต่อไปมันจะทำให้เราเห็นว่า การทำงานปัจจุบันเป็นอย่างไรตรงไหนมีปัญหา ตรงไหนที่มันเป็นคอขวดให้จำไว้ว่า ถ้าขั้นตอนมันห่วยแล้วจะเอาระบบการทำงานแบบอัตโนมัติมาใช้ มันก็ห่วยดังนั้นปรับปรุงการทำงานก่อนนะมันช่วยทำให้เราพูดคุยกัน ทำงานร่วมกันเพื่อเข้าใจซึ่งกันและกัน ได้ช่วยกัน review ขั้นตอนการทำงานอีกด้วยเมื่อถึงขั้นตอนตรงนี้ จะเห็นว่ามันคือเรื่องพื้นฐานมาก ๆ ที่เรามักไม่ทำกัน ใช่ไหมนะ ?

ไม่ได้สนใจเทคโนโลยีหรือเครื่องมือหรือ framework อะไรเลยมันคือเรื่องของการพูดคุย ทำงานร่วมกัน เปิดเผยจากนั้นก็มาดูขั้นตอนการทำงานตั้งแต่ต้นจนจบเพื่อช่วยกันปรับปรุงให้ดีขึ้นเป้าหมายหลักคือ การแก้ไขปัญหาที่ตั้งไว้ตั้งแต่ต้นนั่นเองต่อจากนี้ต่อไป เริ่มนำเครื่องมือและเทคโนโลยีมาใช้แล้ว

ขั้นตอนที่ 3 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 softwareซึ่งเราอาจจะต้องกำหนดการทำงานเหล่านี้ให้ชัดเจนหรือดีที่สุดคือ ช่วยกันสร้างระบบที่ทำงานแบบอัตโนมัติขึ้นมาเพื่อลดงานที่ต้องมีคนเข้าไปยุ่งเกี่ยวให้มากที่สุดเพราะว่า มันคืองานที่ทำซ้ำ ๆ บ่อยมากและที่สำคัญคือ เพื่อลดปัญหาที่เกิดจากคนให้มากที่สุดเรามักจะได้ยินคำว่า work on my machine !!

ขั้นตอนที่ 4 กระบวนการทดสอบแบบอัตโนมัติ

การทดสอบนั้นเป็นหัวใจของการพัฒนา software และ DevOps เลยมันสะท้อนในเรื่องของคุณภาพ software ที่ส่งมอบมากพอสมควรโดยการทดสอบควรต้องเป็นแบบอัตโนมัติให้ได้มากที่สุดเพื่อช่วย validate ว่า software ของเรายังคงทำงานได้อย่างถูกต้องตามที่คาดหวังที่สำคัญต้องทำงานได้รวดเร็วด้วยมิเช่นนั้น ปัญหาต่าง ๆ ก็ยังคงอยู่เสมอ มันน่ากลัวมาก ๆถ้าเราทำการ deploy software บ่อย ๆ แต่ผิดเยอะบ่อย ๆ !!!

มันไม่น่าจะใช่ผลที่เราต้องการใช่ไหม ?

เมื่อเราทำตามมาจนถึงขั้นตอนนี้แล้วเราได้ผ่านความต้องการพื้นฐานตามแนวคิด DevOps มาแล้วสามารถแก้ไขปัญหาที่เรากำหนดได้แล้วจากนี้ก็เป็นเรื่องที่เราต้องไปต่อแล้วไม่ว่าจะเริ่มสร้างระบบอื่น ๆ ขึ้นมา เพื่อให้ใช้งานง่ายขึ้นเช่นระบบแบบ self-service ไม่ต้องมาผ่านขั้นตอนที่มากมายหรือช้ากันแล้วอยากได้อะไรไปสร้างเองได้เลยมันก็อยู่ที่ความต้องการขององค์กรต่อไปแล้วขอเน้นว่า เรื่องพื้นฐานสำคัญมาก ๆ [4]

อ้างอิง

[1] “DevOps คืออะไร - Amazon Web Services (AWS).” Amazon Web Services, Inc., https://aws.amazon.com/th/devops/what-is-devops/. Accessed 26 Feb. 2021.

[2] Chaonithi, Kriangkrai. “DevOps คืออะไร นำมาประโยชน์ได้ยังไง และตัวอย่างการทำ DevOps ที่ Credit OK.” SPICYDOG’s Blog, 16 Nov. 2019, <https://www.spicydog.org/blog/introduction-to-devops-and-the-practical-use-cases-at-credit-ok/>.

[3] DevOps CI/CD คืออะไร? – Ton’s Blog. https://ton.packetlove.com/blog/life-style/devops-ci-cd.html. Accessed 26 Feb. 2021.

[4] แนวทางในการนำ DevOps มาใช้ปรับปรุงการพัฒนา Software. https://www.somkiat.cc/think-before-start-devops/. Accessed 26 Feb. 2021.