(1)DevOps คือ ชุดแนวทางการพัฒนาซอฟต์แวร์ที่รวมการพัฒนาซอฟต์แวร์และการดำเนินงานด้านเทคโนโลยีสารสนเทศ เพื่อลดวงจรชีวิตของการพัฒนาระบบในขณะที่นำเสนอคุณลักษณะการแก้ไขและการอัปเดตบ่อยครั้งเพื่อให้สอดคล้องกับวัตถุประสงค์ทางธุรกิจ DevOps มีหลักการคือ การนำความคล่องตัวมาสู่งานการพัฒนาและการดำเนินงานระบบ DevOps อาจไม่สามารถทดแทน Agile ได้แต่ DevOps จะสามารถนำมาปรับปรุงต่อยอดส่วนที่ยังขาดในเรื่องของการดำเนินงานได้ดี DevOps อาจไม่ใช่การแทนที่ Agile ได้ แต่ก็อาจใช้ร่วมเพื่อให้การปฏิบัติมีระบบดีขึ้น เมื่อเวลาผ่านไป Agile ก็ได้เพิ่มความท้าทายและ DevOps ก็กลายเป็นแนวทางปฏิบัติที่เหมาะสมที่สุด (1)

**CI คืออะไร**  
CI (Continuous Integration)คือ กระบวนการรวม source code ของคนในทีมพัฒนาเข้าด้วยกัน และมีการ test ด้วย test script เพื่อให้แน่ใจว่าไม่มี error ในส่วนใดๆ ของโปรแกรม แล้วถึงทำการ commit ไปที่ branch master อีกต่อนึง  
  
โดยในการพัฒนานั้น มักใช้ Build Server มาช่วย กล่าวคือจะเริ่มทำการ Integration กันตั้งแต่เมื่อมีการเปลี่ยนแปลง Source Code ที่ Repository กลาง ระบบจะทำการตรวจสอบ Code หลังจากการเปลี่ยนแปลงว่าทำงานร่วมกันได้หรือไม่ตั้งแต่ Compile, Testing

**CD คืออะไร**  
1. CD (Continuous Deployment) คือ การ Deploy ขึ้น production โดยจะทำทุกขั้นตอน ตั้งแต่ complie build ไปจนถึง deploy ขึ้น production แบบอัตโนมัติทั้งหมด

2. CD (Continuous Delivery) คือ การทำทุกขั้นตอนคล้ายกันกับ Continuous Deployment ต่างกันตรงที่จะไม่มีการ deploy ขึ้น production ขึ้นในทันที แต่จะเป็นการทำ manual deploy หรือจะเป็นแบบ one click deploy ก็ได้ หลังจาก QA หรือ ฝ่าย Business พอใจในตัว product ที่ทีมทำออกมา (2)

ขั้นตอนที่ 0 ระบุปัญหาก่อนว่ามีอะไรบ้าง ?

จากนั้นจึงทำการเรียกลำดับตามความสำคัญหรือผลกระทบที่เกิดขึ้นว่าจะแก้ไขอะไรก่อนหลังแนะนำให้ค่อย ๆ แก้ไขทีละเรื่องถ้ามันสำคัญหมด คงจะมั่วกันน่าดู เนื่องจากเราจะไร้ทิศทางมาก ๆ ถ้าไม่มีปัญหา จะปรับปรุงไปทำไม จริงไหม ?

ขั้นตอนที่ 1 ลดความหลากหลาย

สิ่งที่พบเจอบ่อยมาก ๆ ในการพัฒนา software คือ ความหลากหลายของการทำงาน ไม่ว่าจะเป็นขั้นตอนการทำงานEnvironment ต่าง ๆ ที่ใช้งานเครื่องมือที่หลายหลาย ตามใจใครหลาย ๆ คนการ configuration ที่ไร้ทิศทางข้อมูลที่หลายหลายชนิด อยู่หลายที่คำถามคือ ถ้ายังมีความหลากหลายแบบนี้แล้วเราจะจัดการอย่างไร ? ยิ่งเริ่มเอาระบบการทำงานแบบอัตโนมัติเข้ามาช่วยปรับปรุง มันยิ่งลำบากหรือไม่ ?

ดังนั้น ควรต้องลดความหลากหลายหรือลองหาสักแนวทางสำหรับการเริ่มต้นดีไหม ?ยิ่งมีความหลากหลาย ยิ่งก่อให้เกิดปัญหาหรือไม่ ?ดังนั้นลองดูว่า อะไรบ้างที่ตัดออกไปแล้วมันทำให้เราทำงานง่ายขึ้นบ้าง ชีวิตน่าจะดีขึ้นนะ

ขั้นตอนที่ 2 ต้องรู้ขั้นตอนการทำงานตั้งแต่ต้นจนจบ

เราไม่สามารถจะปรับปรุงอะไรได้ถ้าเราไม่รู้กระบวนการทำงานตั้งแต่ต้นจนจบเนื่องจากเรามักจะมีคนที่รู้เฉพาะจุดหรือบางส่วนคำถามหรือแล้วภาพใหญ่ละบอกเลยว่า ไม่มีใครรู้หรือรู้ก็แบบลาง ๆ ต้องทำการนัดประชุมกันใหญ่โต หรือใช้เวลานานมาก ๆดังนั้นสิ่งที่ควรทำคืออะไรที่ไม่รู้ทำให้รู้ (มักจะไม่รู้ว่า ไม่รู้อะไร !!) จากนั้นทำการสร้างเอกสาร เพื่ออธิบายการทำงานตั้งแต่ต้นจนจบไว้จากนั้น share ให้ทุกคนที่เกี่ยวข้องจะได้ทำความเข้าใจ แก้ไขหรือปรับปรุงกันต่อไปมันจะทำให้เราเห็นว่า การทำงานปัจจุบันเป็นอย่างไรตรงไหนมีปัญหา ตรงไหนที่มันเป็นคอขวดให้จำไว้ว่า ถ้าขั้นตอนมันห่วยแล้วจะเอาระบบการทำงานแบบอัตโนมัติมาใช้ มันก็ห่วยดังนั้นปรับปรุงการทำงานก่อนนะมันช่วยทำให้เราพูดคุยกัน ทำงานร่วมกันเพื่อเข้าใจซึ่งกันและกัน ได้ช่วยกัน review ขั้นตอนการทำงานอีกด้วยเมื่อถึงขั้นตอนตรงนี้ จะเห็นว่า

มันคือเรื่องพื้นฐานมาก ๆ ที่เรามักไม่ทำกัน ใช่ไหมนะ ?ไม่ได้สนใจเทคโนโลยีหรือเครื่องมือหรือ framework อะไรเลย

มันคือเรื่องของการพูดคุย ทำงานร่วมกัน เปิดเผยจากนั้นก็มาดูขั้นตอนการทำงานตั้งแต่ต้นจนจบเพื่อช่วยกันปรับปรุงให้ดีขึ้น

เป้าหมายหลักคือ การแก้ไขปัญหาที่ตั้งไว้ตั้งแต่ต้นนั่นเองต่อจากนี้ต่อไป เริ่มนำเครื่องมือและเทคโนโลยีมาใช้แล้ว

ขั้นตอนที่ 3 เข้าสู่กระบวนการพัฒนาและส่งมอบ software

การพัฒนาและส่งมอบ software ที่ดีประกอบไปด้วยการจัดการ source code หรือ Source Control Management (SCM)กระบวนการ build ของ softwareกระบวนการ deploy softwareการ 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 ของเรายังคงทำงานได้อย่างถูกต้องตามที่คาดหวัง

ที่สำคัญต้องทำงานได้รวดเร็วด้วยมิเช่นนั้น ปัญหาต่าง ๆ ก็ยังคงอยู่เสมอ มันน่ากลัวมาก ๆถ้าเราทำการ deploy software บ่อย ๆ แต่ผิดเยอะบ่อย ๆ !มันไม่น่าจะใช่ผลที่เราต้องการใช่ไหม ?

เมื่อเราทำตามมาจนถึงขั้นตอนนี้แล้วเราได้ผ่านความต้องการพื้นฐานตามแนวคิด DevOps มาแล้วสามารถแก้ไขปัญหาที่เรากำหนดได้แล้วจากนี้ก็เป็นเรื่องที่เราต้องไปต่อแล้วไม่ว่าจะเริ่มสร้างระบบอื่น ๆ ขึ้นมา เพื่อให้ใช้งานง่ายขึ้นเช่นระบบแบบ self-service ไม่ต้องมาผ่านขั้นตอนที่มากมายหรือช้ากันแล้วอยากได้อะไรไปสร้างเองได้เลย(3)

อ้างอิง

1. อุดมธนะธีระ เ. ความแตกต่างระหว่างการพัฒนาแบบ DevOps กับ Agile [อินเทอร์เน็ต]. iOK2U สารสนเทศเพื่อคุณ. [อ้างถึง 26 กุมภาพันธ์ 2021]. Available at: https://iok2u.com/index.php/article/information-technology/536-devops-agile-2

2. Administrator. การพัฒนาระบบตามแนวทาง CI/CD และ DevOps คืออะไร? [อินเทอร์เน็ต]. [อ้างถึง 26 กุมภาพันธ์ 2021]. Available at: https://www.softmelt.com/article.php?id=664

3. แนวทางในการนำ DevOps มาใช้ปรับปรุงการพัฒนา software [อินเทอร์เน็ต]. [อ้างถึง 26 กุมภาพันธ์ 2021]. Available at: https://www.somkiat.cc/think-before-start-devops/