**DevOps คืออะไร มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร**

DevOps คือการผสมผสานแนวความคิดเชิงวัฒนธรรม แนวทางปฏิบัติ และเครื่องมือต่างๆ ที่ช่วยเพิ่มความสามารถขององค์กรในการส่งมอบแอปพลิเคชันและบริการอย่างรวดเร็ว โดยพัฒนาและปรับปรุงผลิตภัณฑ์ต่างๆ ให้เร็วกว่ากระบวนการการพัฒนาซอฟต์แวร์และการจัดการโครงสร้างพื้นฐานแบบดั้งเดิม ความรวดเร็วนี้ช่วยให้องค์กรสามารถให้บริการแก่ลูกค้าของตนได้ดีขึ้น และสามารถแข่งขันในตลาดได้อย่างมีประสิทธิภาพมากขึ้น

สำหรับโมเดล DevOps ทีมพัฒนาและทีมปฏิบัติการจะไม่ทำงานแบบ “ต่างคนต่างทำ” อีกต่อไป บางครั้ง ทั้งสองทีมจะจับมือร่วมงานเป็นทีมเดียวกันโดยที่เหล่าวิศวกรจะทำงานตลอดวงจรการทำงานของแอปพลิเคชัน ตั้งแต่การพัฒนาและการทดสอบไปจนถึงการปรับใช้และการปฏิบัติงาน และพัฒนาขอบเขตความสามารถที่ไม่ได้จำกัดอยู่ที่การทำงานเพียงอย่างเดียว

ในบางโมเดลของ DevOps ทีมประกันคุณภาพและทีมรักษาความปลอดภัยอาจทำงานรวมกับทีมพัฒนาและทีมปฏิบัติการอย่างใกล้ชิดยิ่งขึ้นตลอดวงจรการทำงานของแอปพลิเคชัน ในเวลาที่การรักษาความปลอดภัยเป็นจุดสำคัญของทุกคนในทีม DevOps เราจะเรียกว่า DevSecOps

ทีมต่างๆ ใช้ข้อปฏิบัติในการเปลี่ยนกระบวนการต่างๆ ที่เคยทำงานแบบแมนนวลและเชื่องช้าให้ทำงานอัตโนมัติ พวกเขาใช้ชุดเทคโนโลยีและเครื่องมือต่างๆ ที่ช่วยให้พวกเขาดำเนินการและพัฒนาแอปพลิเคชันได้อย่างรวดเร็วและเชื่อถือได้ นอกจากนั้น เครื่องมือเหล่านี้ยังช่วยให้วิศวกรเหล่านี้ทำงานได้อย่างอิสระ (เช่น การปรับใช้โค้ด หรือการจัดเตรียมโครงสร้างพื้นฐาน) ที่โดยปกติแล้วจำเป็นต้องได้รับความช่วยเหลือจากทีมอื่น พร้อมทั้งยังช่วยทีมงานทำงานได้รวดเร็วยิ่งขึ้นอีกด้วย

ประโยชน์ของ DevOps

ความรวดเร็ว

ดำเนินการอย่างรวดเร็วเพื่อให้คุณสามารถสร้างสรรค์นวัตกรรมให้แก่ลูกค้าได้รวดเร็วยิ่งขึ้น ปรับตัวต่อการเปลี่ยนแปลงของตลาดได้ดียิ่งขึ้น และมีประสิทธิภาพยิ่งขึ้นในการกระตุ้นให้เกิดผลลัพธ์ทางธุรกิจ โมเดล DevOps จะทำให้นักพัฒนาและทีมปฏิบัติการของคุณบรรลุผลดังกล่าวได้ ตัวอย่างเช่น [ไมโครเซอร์วิส](https://aws.amazon.com/th/devops/what-is-devops/#microservices)และ[การส่งมอบอย่างต่อเนื่อง](https://aws.amazon.com/devops/continuous-delivery/)จะทำให้ทีมมีความเป็นเจ้าของบริการและออกอัปเดตให้แก่ลูกค้าได้รวดเร็วยิ่งขึ้น

การส่งมอบอย่างรวดเร็ว

ออกรุ่นใหม่ๆ ถี่ขึ้นและรวดเร็วยิ่งขึ้น เพื่อให้คุณสามารถสร้างสรรค์นวัตกรรมและปรับปรุงผลิตภัณฑ์ของคุณได้รวดเร็วยิ่งขึ้น ยิ่งคุณสามารถออกคุณสมบัติใหม่ๆ และแก้ไขจุดบกพร่องได้เร็วเท่าไร คุณก็ยิ่งตอบสนองความต้องการของลูกค้าและสร้างความได้เปรียบในการแข่งขันได้เร็วเท่านั้น [การบูรณาการอย่างต่อเนื่อง](https://aws.amazon.com/devops/continuous-integration/)และ[การส่งมอบอย่างต่อเนื่อง](https://aws.amazon.com/devops/continuous-delivery/) เป็นข้อปฏิบัติที่ทำให้กระบวนการออกซอฟต์แวร์ทำงานอัตโนมัติตั้งแต่การสร้างไปจนถึงการปรับใช้

ความเชื่อถือได้

ทำให้แน่ใจถึงคุณภาพของการอัปเดตแอปพลิเคชันและการเปลี่ยนแปลงโครงสร้างพื้นฐาน เพื่อให้คุณสามารถส่งมอบได้อย่างน่าเชื่อถือในความเร็วที่มากขึ้น ในขณะที่ยังคงรักษาประสบการณ์ใช้งานเชิงบวกของผู้ใช้ปลายทาง ใช้ข้อปฏิบัติอย่างเช่น [การบูรณาการอย่างต่อเนื่อง](https://aws.amazon.com/devops/continuous-integration/) และ[การส่งมอบอย่างต่อเนื่อง](https://aws.amazon.com/devops/continuous-delivery/) เพื่อทดสอบว่าการเปลี่ยนแปลงนั้นทำงานได้และปลอดภัยหรือไม่ ข้อปฏิบัติใน[การตรวจสอบและการบันทึก](https://aws.amazon.com/th/devops/what-is-devops/" \l "monitoring)จะช่วยให้คุณทราบถึงประสิทธิภาพการทำงานในทันที

ขนาด

ดำเนินการและจัดการโครงสร้างพื้นฐานและกระบวนการพัฒนาของคุณในขนาดต่างๆ การทำงานอัตโนมัติและความสอดคล้องจะช่วยให้คุณจัดการระบบที่ซับซ้อนหรือมีการเปลี่ยนแปลงได้อย่างมีประสิทธิภาพ โดยมีความเสี่ยงที่ลดลง ตัวอย่างเช่น [โครงสร้างพื้นฐานเป็นโค้ด](https://aws.amazon.com/th/devops/what-is-devops/#iac)จะช่วยคุณในการจัดการสภาพแวดล้อมในการพัฒนา การทดสอบ และการทำงานจริงในลักษณะที่ทำซ้ำได้และมีประสิทธิภาพยิ่งขึ้น

การทำงานร่วมกันที่ปรับปรุงดีขึ้น

สร้างทีมงานที่มีประสิทธิผลมากขึ้นภายใต้โมเดลเชิงวัฒนธรรมของ DevOps ซึ่งให้ความสำคัญกับคุณ เช่น ความเป็นเจ้าของและความรับผิดชอบ นักพัฒนาและทีมปฏิบัติการจะ[ทำงานร่วมกัน](https://aws.amazon.com/th/devops/what-is-devops/" \l "communication)อย่างใกล้ชิด รับผิดชอบร่วมกันในหลายกรณี และนำลำดับการทำงานมารวมเข้าด้วยกัน ซึ่งจะช่วยลดความไร้ประสิทธิภาพและประหยัดเวลา (เช่น ลดระยะเวลาการส่งมอบระหว่างนักพัฒนากับทีมปฏิบัติการ เขียนโค้ดที่คำนึงถึงสภาพแวดล้อมที่นำโค้ดไปใช้งาน)

การรักษาความปลอดภัย

ดำเนินการอย่างรวดเร็วในขณะที่ยังคงรักษาการควบคุมและการปฏิบัติสอดคล้องกับกฎระเบียบ คุณสามารถรับเอาโมเดล DevOps มาใช้โดยไม่ต้องสูญเสียความปลอดภัยโดยใช้นโยบายการปฏิบัติตามกฎระเบียบที่ทำงานอัตโนมัติ การควบคุมโดยละเอียด และเทคนิคการจัดการการกำหนดค่า ตัวอย่างเช่น คุณสามารถใช้โครงสร้างพื้นฐานเป็นโค้ดและ[นโยบายเป็นโค้ด](https://aws.amazon.com/th/devops/what-is-devops/" \l "policyascode) เพื่อกำหนดและติดตามการปฏิบัติตามกฎระเบียบในขนาดต่างๆ

**CI/CD คืออะไร มีความสัมพันธ์กับ DevOps อย่างไร**

**CI คือ**  
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 คือ**  
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

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 ซึ่งสมัยนี้ก็มี tools ต่างๆ ที่ทำหน้าที่พวกนี้เยอะมากทั้ง On-Premise และ On-Cloud ที่เรารู้จักกันดี ก็น่าจะเป็น Jenkins ที่เข้ามามีบทบาทมากในการทำ CI/CD

**ตัวอย่างขั้นตอนการพัฒนาระบบตามแนวทาง CI/CD**  
1. Developer เมื่อทำการพัฒนา feature เสร็จ จะทำการ build, test และ run บนเครื่องของตัวเอง (Local) เพื่อทำให้แน่ใจว่าระบบทำงานได้ถูกต้องและให้แน่ใจว่าสิ่งที่เปลี่ยนแปลงไม่กระทบส่วนอื่น ๆ  
  
2. ทำการดึง source code ล่าสุดจาก Repository ของระบบ เพื่อตรวจสอบว่ามีการเปลี่ยนแปลงหรือไม่ถ้ามีการเปลี่ยนแปลงก็ให้ทำการรวมหรือ merge ที่เครื่องของ Devleoper ก่อน จากนั้นจึงทำการ build, test และ run อีกรอบ เมื่อทุกอย่างผ่านทั้งหมด ให้ทำการส่งการเปลี่ยนแปลงไปยัง Repository กลาง  
  
3. เมื่อ Repository กลางมีการเปลี่ยนแปลง จะต้องมีระบบ CI ทำการ build หลังจาก build จะส่งต่อไป run unit testing ก่อนถ้าผ่านหมดถึงจะส่งต่อไปยังระบบ Continuous Delivery เพื่อ deploy to sit environment  
  
4. เมื่อ source code ถูก deploy to sit environment แล้วจะ trigger ไปสั่งให้ run job automated testing ใน level ของเทสเคส ซึ่งเป็นชุดเทสเคสย่อยๆ ไม่เยอะมากเฉพาะในส่วนของ feature code ที่ถูก deploy มาเท่านั้น  
  
5. หลังจาก run test เสร็จแล้วถ้าเกิดว่า run มีบางส่วนไม่ผ่านทั้งหมดจะไม่ส่งต่อไปยังระบบ Continuous Delivery เพื่อ deploy to uat environment QA จะทำการ investigate ว่าเกิดจากอะไร เป็นที่ระบบมี Bug เกิดขึ้นจริงหรือไม่ ถ้ามี bug ก็ให้ dev แก้ไข และ deploy มาใหม่ วน loop ใหม่  
  
6. กรณีหลังจาก run test ผ่านทั้งหมดจะส่งต่อไปยังระบบ Continuous Delivery เพื่อ deploy to uat(staging) environment เมื่อ source code ถูก deploy to uat(staging) แล้ว จะ trigger ไปสั่งให้ run job automated testing ใน level ของเทสเคส regression test และ QA ก็ทำการทดสอบ Acceptance testing ไปด้วยพร้อมๆ กันที่ uat(staging) environment นี้ เมื่อมีการ deploy ซำ้ๆ เพื่อ fixed bug จากที่ QA เจอ หรือที่พบเจอจากการ run regression test แล้ว fail ก็จะเป็นการวน loop ตั้งแต่ต้นจนจบ จนกระทั้ง ทุกอย่างผ่านหมด Business ฟันธง มาว่าเอาขึ้น production ได้ เป็นการ confirm ว่าเราจะเอา code version สุดท้ายนี้ขึ้นไปที่ production environment  
  
**ข้อดีของการพัฒนาระบบตามแนวทาง CI/CD**  
ลดระยะเวลาในการพัฒนา ลดต้นทุนในการพัฒนา รองรับการเปลี่ยนแปลงแก้ไขได้ง่าย ดูแลรักษาได้ง่าย เนื่องจากมีการดำเนินไปอย่างต่อเนื่อง เป็นไปอย่างอัตโนมัติและทราบผลลัพธ์ได้ทันที ทำให้เราสามารถส่งมอบแอพฯ รุ่นใหม่ให้กับลูกค้าได้ไวขึ้น

อ้างอิง

1. DevOpsคืออะไร (ออนไลน์). (2564). สืบค้นจาก : http:// https://aws.amazon.com/th/devops/what-is-devops/ [26 กุมภาพันธ์ 2564]

2. การพัฒนาระบบตามแนวทาง CI/CD และ DevOps คืออะไร(ออนไลน์). (2554). สืบค้นจาก : https://www.softmelt.com/article.php?id=664/ [26 กุมภาพันธ์ 2564]