

# DevOps และ CI/CD

## DevOps (Development Operation)

DevOps (Development Operation) คือการผสมผสานแนวความคิดเชิงวัฒนธรรม แนวทางปฏิบัติ และเครื่องมือต่างๆ ที่ช่วยเพิ่มความสามารถขององค์กรในการส่งมอบแอปพลิเคชันและบริการอย่างรวดเร็ว โดยพัฒนาและปรับปรุงผลิตภัณฑ์ต่างๆ ให้เร็วกว่ากระบวนการการพัฒนาซอฟต์แวร์และการจัดการโครงสร้างพื้นฐานแบบดั้งเดิม ความรวดเร็วนี้ช่วยให้องค์กรสามารถให้บริการแก่ลูกค้าของตนได้ดีขึ้น และสามารถแข่งขันในตลาดได้อย่างมีประสิทธิภาพมากขึ้น

สำหรับโมเดล DevOps ทีมพัฒนาและทีมปฏิบัติการจะไม่ทำงานแบบ “ต่างคนต่างทำ” อีกต่อไป บางครั้ง ทั้งสองทีมจะจับมือร่วมงานเป็นทีมเดียวกันโดยที่เหล่าวิศวกรจะทำงานตลอดวงจรการทำงานของแอปพลิเคชัน ตั้งแต่การพัฒนาและการทดสอบไปจนถึงการปรับใช้และการปฏิบัติงาน และพัฒนาขอบเขตความสามารถที่ไม่ได้จำกัดอยู่แค่การทำงานเพียงอย่างเดียว ในบางโมเดลของ DevOps ทีมประกันคุณภาพและทีมรักษาความปลอดภัยอาจทำงานร่วมกับทีมพัฒนาและทีมปฏิบัติการอย่างใกล้ชิดยิ่งขึ้นตลอดวงจรการทำงานของแอปพลิเคชัน ในเวลาที่มีการรักษาความปลอดภัยเป็นจุดสำคัญของทุกคนในทีม DevOps เราจะเรียกว่า DevSecOps ทีมต่างๆ ใช้ข้อปฏิบัติในการเปลี่ยนกระบวนการต่างๆ ที่เคยทำงานแบบแมนนวลและเชื่อมต่อให้ทำงานอัตโนมัติ พวกเขาใช้ชุดเทคโนโลยีและเครื่องมือต่างๆ ที่ช่วยให้พวกเขาดำเนินการและพัฒนาแอปพลิเคชันได้อย่างรวดเร็วและเชื่อถือได้ นอกจากนี้ เครื่องมือเหล่านี้ยังช่วยให้วิศวกรเหล่านี้ทำงานได้อย่างอิสระ (เช่น การปรับใช้โค้ดหรือการจัดเตรียมโครงสร้างพื้นฐาน) ที่โดยปกติแล้วจำเป็นต้องได้รับความช่วยเหลือจากทีมอื่น พร้อมทั้งยังช่วยทีมงานทำงานได้รวดเร็วยิ่งขึ้นอีกด้วย

## ประโยชน์ของ DevOps

**ความรวดเร็ว** ดำเนินการอย่างรวดเร็วเพื่อให้คุณสามารถสร้างสรรค์นวัตกรรมให้แก่ลูกค้าได้รวดเร็วยิ่งขึ้น ปรับตัวต่อการเปลี่ยนแปลงของตลาดได้ดียิ่งขึ้น และมีประสิทธิภาพยิ่งขึ้นในการกระตุ้นให้เกิดผลลัพธ์ทางธุรกิจ โมเดล DevOps จะทำให้นักพัฒนาและทีมปฏิบัติการของคุณบรรลุผลดังกล่าวได้ ตัวอย่างเช่น ไมโครเซอร์วิสและการส่งมอบอย่างต่อเนื่องจะทำให้ทีมมีความเป็นเจ้าของบริการและออกอัปเดตให้แก่ลูกค้าได้รวดเร็วยิ่งขึ้น

**การส่งมอบอย่างรวดเร็ว** ออกรุ่นใหม่ๆ ถีขึ้นและรวดเร็วยิ่งขึ้น เพื่อให้คุณสามารถสร้างสรรค์นวัตกรรมและปรับปรุงผลิตภัณฑ์ของคุณได้รวดเร็วยิ่งขึ้น ยิ่งคุณสามารถออกคุณสมบัติใหม่ๆ และแก้ไขจุดบกพร่องได้เร็วเท่าไร คุณก็ยิ่งตอบสนองความต้องการของลูกค้าและสร้างความได้เปรียบในการแข่งขันได้เร็วเท่านั้น การบูรณาการอย่างต่อเนื่องและการส่งมอบอย่างต่อเนื่อง เป็นข้อปฏิบัติที่ทำให้กระบวนการออกซอฟต์แวร์ทำงานอัตโนมัติตั้งแต่การสร้างไปจนถึงการปรับใช้

**ความเชื่อถือได้** ทำให้แน่ใจถึงคุณภาพของการอัปเดตแอปพลิเคชันและการเปลี่ยนแปลงโครงสร้างพื้นฐาน เพื่อให้คุณสามารถส่งมอบได้อย่างน่าเชื่อถือในความเร็วที่มากขึ้น ในขณะที่ยังคงรักษาประสบการณ์ใช้งานเชิงบวกของผู้ใช้ปลายทาง ใช้ข้อปฏิบัติอย่างเช่น การบูรณาการอย่างต่อเนื่อง และการส่งมอบอย่างต่อเนื่อง เพื่อทดสอบว่าการเปลี่ยนแปลงนั้นทำงานได้และปลอดภัยหรือไม่ ข้อปฏิบัติในการตรวจสอบและการบันทึกจะช่วยให้คุณทราบถึงประสิทธิภาพการทำงานในทันที

**ขนาด** ดำเนินการและจัดการโครงสร้างพื้นฐานและกระบวนการพัฒนาของคุณในขนาดต่างๆ การทำงานอัตโนมัติและความสอดคล้องจะช่วยให้คุณจัดการระบบที่ซับซ้อนหรือมีการเปลี่ยนแปลงได้อย่างมีประสิทธิภาพ โดยมีความเสี่ยงที่ลดลง ตัวอย่างเช่น โครงสร้างพื้นฐานเป็นโค้ดจะช่วยให้คุณในการจัดการสภาพแวดล้อมในการพัฒนา การทดสอบ และการทำงานจริงในลักษณะที่ทำซ้ำได้และมีประสิทธิภาพยิ่งขึ้น

**การทำงานร่วมกันที่ปรับปรุงดีขึ้น** สร้างทีมงานที่มีประสิทธิผลมากขึ้นภายใต้โมเดลเชิงวัฒนธรรมของ DevOps ซึ่งให้ความสำคัญกับคุณ เช่น ความเป็นเจ้าของและความรับผิดชอบ นักพัฒนาและทีมปฏิบัติการจะทำงานร่วมกันอย่างใกล้ชิด รับผิดชอบร่วมกันในหลายกรณี และนำลำดับการทำงานมารวมเข้าด้วยกัน ซึ่งจะช่วยลดความไร้ประสิทธิภาพและประหยัดเวลา (เช่น ลดระยะเวลาการส่งมอบระหว่างนักพัฒนากับทีมปฏิบัติการ เขียนโค้ดที่คำนึงถึงสภาพแวดล้อมที่นำโค้ดไปใช้งาน)

**การรักษาความปลอดภัย** ดำเนินการอย่างรวดเร็วในขณะที่ยังคงรักษาการควบคุมและการปฏิบัติสอดคล้องกับกฎระเบียบ คุณสามารถรับเอาโมเดล DevOps มาใช้โดยไม่ต้องสูญเสียความปลอดภัยโดยใช้นโยบายการปฏิบัติตามกฎระเบียบที่ทำงานอัตโนมัติ การควบคุมโดยละเอียด และเทคนิคการจัดการการกำหนดค่า ตัวอย่างเช่น คุณสามารถใช้โครงสร้างพื้นฐานเป็นโค้ดและนโยบายเป็นโค้ด เพื่อกำหนดและติดตามการปฏิบัติตามกฎระเบียบในขนาดต่างๆ

DevOps เป็นการขจัดกำแพงกันระหว่างทีมพัฒนาและทีมปฏิบัติการที่ปกติต่างคนต่างทำงาน ในบางองค์กร อาจไม่มีการแบ่งแยกทีมพัฒนาและทีมปฏิบัติการเลย วิศวกรอาจทำทั้งสองงาน สำหรับ DevOps ทั้งสองทีมจะทำงานด้วยกันเพื่อปรับให้อัตราผลผลิตของนักพัฒนาและความเชื่อถือได้ของทีมปฏิบัติการอยู่ในระดับที่เหมาะสม ทั้งสองทีมจะต้องพยายามสื่อสารกันบ่อยครั้ง เพื่อเพิ่มประสิทธิภาพ และปรับปรุงคุณภาพของบริการที่มอบให้แก่ลูกค้า พวกเขาจะมีความเป็นเจ้าของบริการของตนอย่างเต็มที่ ซึ่งมักเกินบทบาทหรือตำแหน่งของตนที่ระบุไว้ในขอบเขตที่กำหนดไว้ในแบบเดิมๆ โดยคำนึงถึงความต้องการของลูกค้าและวิธีที่สามารถมีส่วนร่วมในการสนองความต้องการเหล่านั้น ทีมประกันคุณภาพและทีมรักษาความปลอดภัยยังอาจบูรณาการอย่างใกล้ชิดกับทีมดังกล่าวอีกด้วย โดยไม่คำนึงถึงโครงสร้างองค์กร องค์กรที่ใช้โมเดล DevOps จะมีทีมงานที่คอยตรวจสอบวงจรการพัฒนาและโครงสร้างพื้นฐานทั้งหมดโดยเป็นส่วนหนึ่งของความรับผิดชอบของตน

ซอฟต์แวร์และอินเทอร์เน็ตได้เปลี่ยนโลกและอุตสาหกรรมต่างๆ ตั้งแต่การช้อปปิ้งและความบันเทิงไปจนถึงการธนาคาร ซอฟต์แวร์ไม่ได้เพียงแค่สนับสนุนธุรกิจเท่านั้น แต่ยังเป็นองค์ประกอบสำคัญในทุกภาคส่วนของธุรกิจ บริษัทต่างๆ มีปฏิสัมพันธ์กับลูกค้าของตนผ่านทางซอฟต์แวร์ที่ให้บริการหรือมีแอปพลิเคชันทางออนไลน์บนอุปกรณ์ทุกชนิด นอกจากนั้นยังใช้ซอฟต์แวร์เพื่อเพิ่มประสิทธิภาพการทำงานโดยเปลี่ยนแปลงทุกส่วนของห่วงโซ่คุณค่า เช่น โลจิสติกส์ การสื่อสาร และการปฏิบัติการ ในทำนองเดียวกันกับที่บริษัทขายสินค้าที่จับต้องได้เปลี่ยนวิธีการออกแบบ สร้าง และส่งมอบผลิตภัณฑ์โดยใช้การทำงานอัตโนมัติทางอุตสาหกรรมตลอดศตวรรษที่ 20 บริษัทในยุคปัจจุบันก็ต้องเปลี่ยนวิธีการสร้างและส่งมอบซอฟต์แวร์เช่นกัน

## CI/CD (Continuous Integration / Continuous Delivery และ Continuous Deployment)

Continuous Integration (CI) คือ กระบวนการที่ใช้สำหรับการรวบรวมซอฟต์แวร์ที่มีการพัฒนาแยกส่วนกันอย่างต่อเนื่อง อาจจะโดยหนึ่งหรือหลายนักพัฒนาที่ตามที สุกท้ายแล้วซอฟต์แวร์ที่พัฒนาขึ้นเล็กๆ ที่พัฒนาขึ้นมาจะต้องนำมารวมกันเป็นชิ้นใหญ่หนึ่งชิ้น จะทำอะไรให้มั่นใจได้ว่า 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

Continuous Deployment คือ ในทุกๆ ขั้นตอนจนถึงการ deployment ขึ้น production จะทำแบบอัตโนมัติทั้งหมด

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

### ตัวอย่างขั้นตอนการพัฒนาในระบบตามแนวทาง CI/CD

1. Developer เมื่อทำการพัฒนา feature เสร็จ จะทำการ build, test และ run บนเครื่องของตัวเอง (Local) เพื่อให้แน่ใจว่าระบบทำงานได้ถูกต้องและให้แน่ใจว่าสิ่งที่เปลี่ยนแปลงไม่กระทบส่วนอื่น ๆ
2. ทำการดึง source code ล่าสุดจาก Repository ของระบบ เพื่อตรวจสอบว่ามีการเปลี่ยนแปลงหรือไม่ถ้ามีการเปลี่ยนแปลงก็ให้ทำการรวมหรือ merge ที่เครื่องของ Developer ก่อน จากนั้นจึงทำการ build, test และ run อีกรอบ เมื่อทุกอย่างผ่านทั้งหมด ให้ทำการส่งการเปลี่ยนแปลงไปยัง Repository กลาง
3. เมื่อ Repository กลางมีการเปลี่ยนแปลง จะต้องมีการ 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 ของทดสอบ ซึ่งเป็นชุดทดสอบย่อยๆ ไม่เยอะมากเฉพาะในส่วน 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 ของทดสอบ regression test และ QA ก็ทำการทดสอบ Acceptance testing ไปด้วยพร้อมๆ กันที่ uat(staging) environment นี้ เมื่อมีการ deploy ซ้ำๆ เพื่อ fixed bug จากที่ QA เจอ หรือที่พบเจอจากการ run regression test แล้ว fail ก็จะเป็นการวน loop ตั้งแต่ต้นจนจบ จนกระทั่ง ทุกอย่างผ่านหมด Business พิจารณาว่าเอาขึ้น production ได้ เป็นการ confirm ว่าเราจะเอา code version สุดท้ายนี้ขึ้นไป production environment

#### **ข้อดีของการพัฒนาระบบตามแนวทาง CI/CD**

ลดระยะเวลาในการพัฒนา ลดต้นทุนในการพัฒนา รองรับการเปลี่ยนแปลงแก้ไขได้ง่าย ดูแลรักษาได้ง่าย เนื่องจากมีการดำเนินไปอย่างต่อเนื่อง เป็นไปอย่างอัตโนมัติและทราบผลลัพธ์ได้ทันที ทำให้เราสามารถส่งมอบแอปฯ รุ่นใหม่ให้กับลูกค้าได้ไวขึ้น

## อ้างอิง

AWS. 2562. DevOps คืออะไร. (ออนไลน์). แหล่งที่มา : <https://aws.amazon.com/th/devops/what-is-devops/>

Narissara. 2562. การนำ Automated testing มารวมเข้ากับ กระบวนการ CI/CD. (ออนไลน์). แหล่งที่มา : <https://medium.com/@narissara.tan>