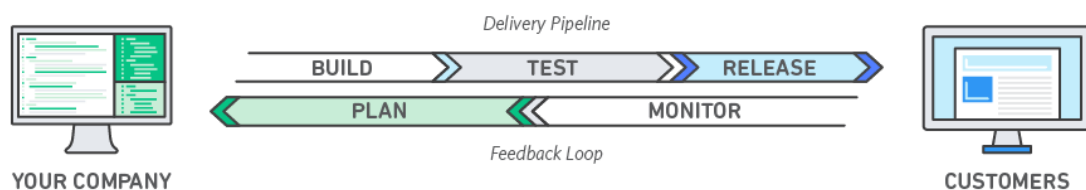


## สรุปเนื้อหาเกี่ยวกับ DevOps และ CI/CD

Q : DevOps คืออะไร มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร

DevOps คือการผสมผสานแนวความคิดเชิงวัฒนธรรม แนวทางปฏิบัติ และเครื่องมือต่างๆ ที่ช่วยเพิ่มความสามารถขององค์กรในการส่งมอบแอปพลิเคชันและบริการอย่างรวดเร็ว โดยพัฒนาและปรับปรุงผลิตภัณฑ์ต่างๆ ให้เร็วกว่ากระบวนการการพัฒนาซอฟต์แวร์และการจัดการโครงสร้างพื้นฐานแบบดั้งเดิม ความรวดเร็วนี้ช่วยให้องค์กรสามารถให้บริการแก่ลูกค้าของตนได้ดีขึ้น และสามารถแข่งขันในตลาดได้อย่างมีประสิทธิภาพมากขึ้น



รูปที่ 1 แสดงการทำงานของ DevOps ระหว่างบริษัทกับลูกค้า

### DevOps ทำงานอย่างไร

สำหรับโมเดล DevOps ทีมพัฒนาและทีมปฏิบัติการจะไม่ทำงานแบบ “ต่างคนต่างทำ” อีกต่อไป บางครั้ง ทั้งสองทีมจะจับมือร่วมงานเป็นทีมเดียวกันโดยที่เหล่าวิศวกรจะทำงานตลอดวงจรการทำงานของแอปพลิเคชัน ตั้งแต่การพัฒนาและการทดสอบไปจนถึงการปรับใช้และการปฏิบัติงาน และพัฒนาขอบเขตความสามารถที่ไม่ได้จำกัดอยู่ที่การทำงานเพียงอย่างเดียว

ในบางโมเดลของ DevOps ทีมประกันคุณภาพและทีมรักษาความปลอดภัยอาจทำงานร่วมกับทีมพัฒนาและทีมปฏิบัติการอย่างใกล้ชิดยิ่งขึ้นตลอดวงจรการทำงานของแอปพลิเคชัน ในเวลาที่มีการรักษาความปลอดภัยเป็นจุดสำคัญของทุกคนในทีม DevOps เราจะเรียกว่า DevSecOps

ทีมต่างๆ ใช้ข้อปฏิบัติในการเปลี่ยนกระบวนการต่างๆ ที่เคยทำงานแบบแมนนวลและเชื่อมต่อให้ทำงานอัตโนมัติ พวกเขาใช้ชุดเทคโนโลยีและเครื่องมือต่างๆ ที่ช่วยให้พวกเขาดำเนินการและพัฒนาแอปพลิเคชันได้อย่างรวดเร็วและเชื่อถือได้นอกจากนั้น เครื่องมือเหล่านี้ยังช่วยให้วิศวกรเหล่านี้ทำงานได้อย่างอิสระ (เช่น การปรับใช้โค้ด หรือการจัดเตรียมโครงสร้างพื้นฐาน) ที่โดยปกติแล้วจำเป็นต้องได้รับความช่วยเหลือจากทีมอื่น พร้อมทั้งยังช่วยทีมงานทำงานได้รวดเร็วยิ่งขึ้น

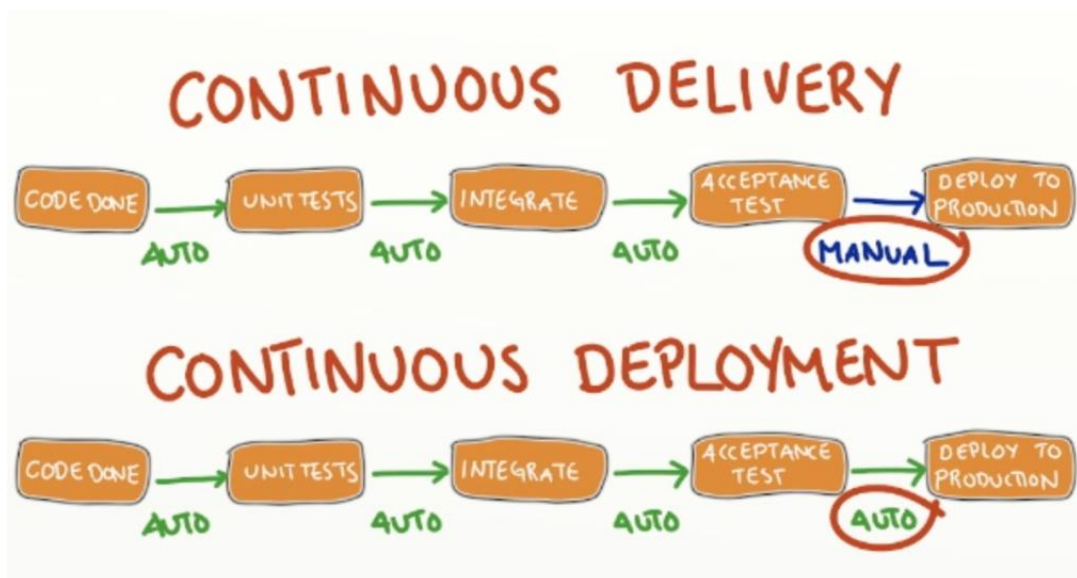
Q : CI/CD คืออะไร มีความสัมพันธ์กับ DevOps อย่างไร

CI/CD คืออะไร ?

Continuous Integration (CI) คือ กระบวนการรวม source code ของคนในทีมพัฒนาเข้าด้วยกัน และมีการ test ด้วย test script เพื่อให้แน่ใจว่าไม่มี error ในส่วนใดๆ ของโปรแกรม แล้วถึงทำการ commit ไปที่ branch master อีกต่อไป

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 ก็ได้



รูปที่ 2 แสดงแผนภาพความสัมพันธ์ระหว่าง CI กับ CD

ดังนั้น ความสัมพันธ์ระหว่าง DevOps กับ CI/CD คือ DevOps จะเป็นผู้พัฒนาโดยอาศัยหลักการของ CI/CD ของการทำ Software ซึ่งนำมาสู่การผลิตพันธ์ เพื่อส่งมอบให้กับลูกค้า

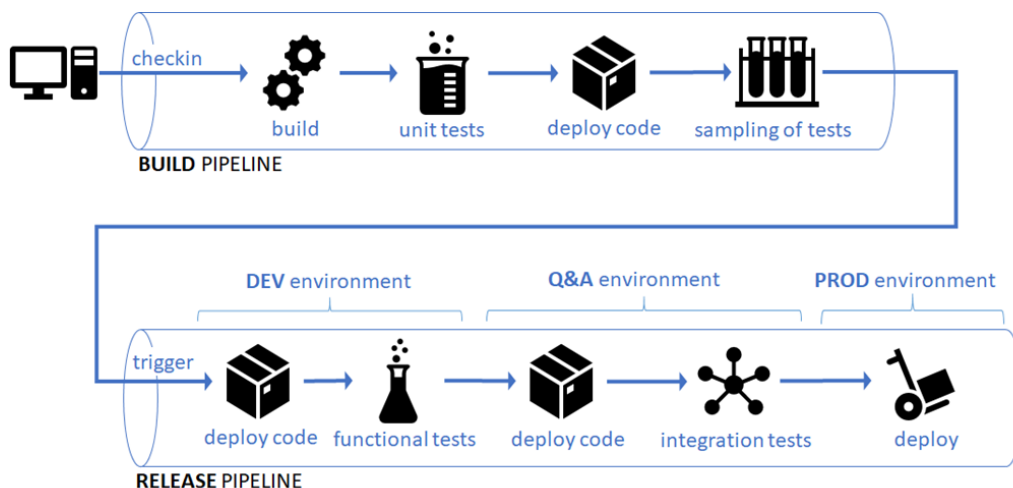
Q : หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์จะต้องทำอย่างไรบ้าง

การนำเอา CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์ ยกตัวอย่างในส่วนของการทำ Automated test ซึ่ง Automated testing ถูกนำไปใช้ใน Process ของ CI

กล่าวคือ ทุกการเปลี่ยนแปลงหนึ่งนั้นคือการทดสอบนั่นเอง แน่นอนว่า ต้องเป็นการทดสอบแบบอัตโนมัติอย่างแน่นอน (Automated testing) ไม่ว่าจะเป็น Unit testing, Integration testing และ Functional testing

ทางทีมพัฒนาระบบของเราจะแบ่ง environment เป็น 4 ส่วนคือ Local , Alpha , Staging , Production มีขั้นตอนดังนี้

1. Developer เมื่อทำการพัฒนา feature เสร็จ จะทำการ build, test และ run บนเครื่องของตัวเอง (Local) เพื่อให้แน่ใจว่าระบบทำงานได้ถูกต้องและให้แน่ใจว่าสิ่งที่เปลี่ยนแปลงไม่กระทบส่วนอื่น ๆ
2. ทำการดึง source code ล่าสุดจาก Repository ของระบบ เพื่อตรวจสอบว่ามีการเปลี่ยนแปลงหรือไม่ถ้ามีการเปลี่ยนแปลงก็ให้ทำการรวม หรือ merge ที่เครื่องของ Developer ก่อน จากนั้นจึงทำการ build, test และ run อีกรอบเมื่อทุกอย่างผ่านทั้งหมด ให้ทำการส่งการเปลี่ยนแปลงไปยัง Repository กลาง
3. เมื่อ Repository กลางมีการเปลี่ยนแปลง จะต้องมีการ CI ทำการ build, หลังจาก build จะส่งต่อไป run unit testing ก่อนถ้าผ่านหมดถึงจะส่งต่อไปยังระบบ Continuous Delivery เพื่อ deploy to alpha environment



รูปที่ 3 แผนภาพแสดงการทำงานในAutomate Test ส่วนของ CI/CD

4. เมื่อ source code ถูก deploy to alpha environment แล้วจะ trigger ไปสั่งให้ run job automated testing ใน level ของทดสอบ smoke test ซึ่งเป็นชุดทดสอบย่อยๆไม่เยอะมากเฉพาะในส่วน of feature code ที่ถูก deploy มาเท่านั้น

5. หลังจาก run smoke test เสร็จแล้วถ้าเกิดว่า run มีบางส่วนไม่ผ่านทั้งหมดจะไม่ส่งต่อไปยังระบบ Continuous Delivery เพื่อ deploy to staging environment QA จะทำการ investigate ว่าเกิดจากอะไร เป็นที่ระบบมี Bug เกิดขึ้นจริงหรือไม่ ถ้ามี bug ก็ให้ dev แก้ไข และ deploy มาใหม่ วน loop ใหม่

6. กรณีหลังจาก run smoke test ผ่านทั้งหมดจะส่งต่อไปยังระบบ Continuous Delivery เพื่อ deploy to Staging environment เมื่อ source code ถูก deploy to staging แล้ว จะ trigger ไปสั่งให้ run job automated testing ใน level ของทดสอบ regression test และ QA ก็ทำการทดสอบ Acceptance testing ไปด้วยพร้อมๆกันที่ Staging environment นี้ เมื่อมีการ deploy ซ้ำๆ เพื่อ fix bug จากที่ QA เจอ หรือที่พบเจอจากการ run regression test แล้ว fail ก็จะเป็นการวน loop ตั้งแต่ต้นจนจบ จนกระทั่ง ทุกอย่างผ่านหมด Business ฟันธง!! มาว่าเอาขึ้น production ได้ เป็นการ confirm ว่าเราจะเอา code version สุดท้ายนี้ขึ้นไปที่ production environment

แนวทางในการนำ DevOps มาใช้ปรับปรุงการพัฒนา software มีขั้นตอนดังนี้

ขั้นตอนที่ 1 ระบุปัญหาว่ามีอะไรบ้าง

ขั้นตอนที่ 2 ลดความหลากหลาย เพื่อให้การพัฒนา Software มีการใช้ภาษาเดียวกัน และใช้งานเป็นไปในรูปแบบเดียวกันมากขึ้น

ขั้นตอนที่ 3 ต้องรู้ขั้นตอนการทำงานตั้งแต่ต้นจนจบ

ขั้นตอนที่ 4 เข้าสู่กระบวนการพัฒนาและส่งมอบ software

ขั้นตอนที่ 5 กระบวนการทดสอบแบบอัตโนมัติ

## เอกสารอ้างอิง

amazon.com. (2564). DevOps คืออะไร. ค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก <https://aws.amazon.com/th/devops/what-is-devops/>

Thraithep Junthep. (2561). เมื่อหัดทำ CI/CD ครั้งแรก. ค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก <https://www.medium.com/@thraithep.a/เมื่อหัดทำ-ci-cd-ครั้งแรก-f4f79151311c>

Narissara. (2560). การนำ Automated testing มารวมเข้ากับ กระบวนการ CI/CD. ค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก [medium.com/@narissara.tan/การนำ-automated-testing-มารวมเข้ากับ-กระบวนการ-ci-cd-58c97e31e230](https://medium.com/@narissara.tan/การนำ-automated-testing-มารวมเข้ากับ-กระบวนการ-ci-cd-58c97e31e230)

Somkiat. (2563). แนวทางในการนำ DevOps มาใช้ปรับปรุงการพัฒนา software. ค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก <https://www.somkiat.cc/think-before-start-devops/>