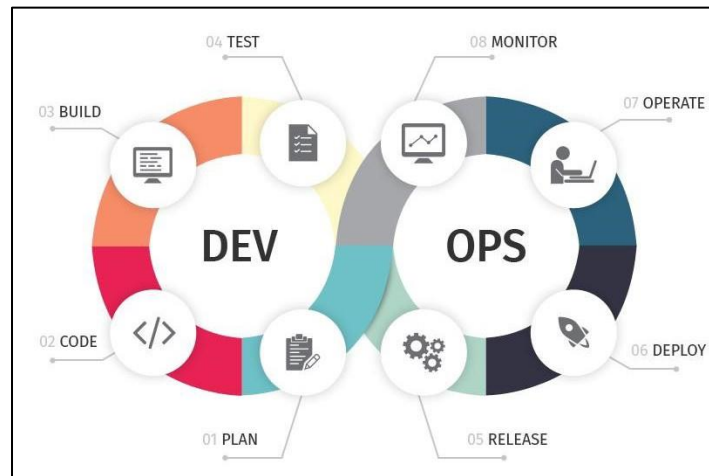


## 1. DevOps คืออะไร มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร



ที่มา: <https://medium.com/lindevth/5-คุณลักษณะสำคัญของ-devops-1b4df97ed70d>

DevOps คือ การนำเอา Development กับ Operations มารวมกัน โดยจุดเริ่มต้นของ DevOps เกิดขึ้นจากความขัดแย้งของ Development กับ Operations โดยปกติแล้ว Development ทำหน้าที่พัฒนาระบบให้ทันการเปลี่ยนแปลงของตลาดอย่างรวดเร็ว แต่ Operations ทำหน้าที่บริหารระบบที่มีความเสถียร มั่นคง เชื่อถือได้ให้กับลูกค้า การทำให้ทั้งสองบรรลุเป้าหมายในเวลาเดียวกันถือว่าเป็นเรื่องที่ยากมาก เพราะ หน่วยงาน IT ไม่สามารถเปลี่ยนระบบด้วยความรวดเร็วในขณะที่ทำให้ระบบเสถียรไปพร้อมกันได้ เพราะการเปลี่ยนระบบแต่ละครั้งอาจจะทำให้เกิดอันตรายต่อระบบได้

DevOps คือ รูปแบบและวิธีการปฏิบัติ วัฒนธรรม และกระบวนการต่างๆ เพื่อแก้ไขปัญหาที่เกิดจากความขัดแย้งระหว่าง Development และ Operations และ ช่วยเพิ่มประสิทธิภาพในการทำงานให้สามารถผลิต software ออกสู่ตลาดได้อย่างรวดเร็ว มีคุณภาพและเสถียรภาพมากขึ้น ในขณะที่ ราคา ลดลง เพื่อให้บริษัทสำเร็จตามที่วางเป้าหมายไว้

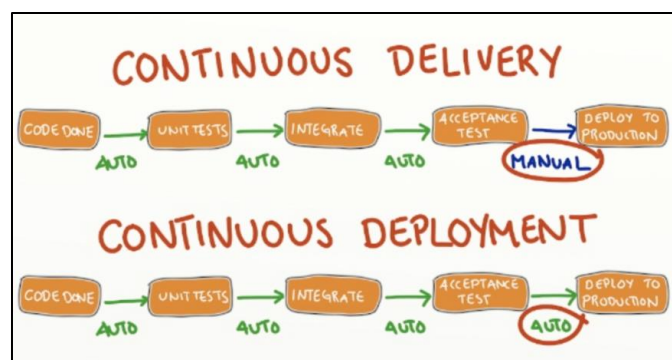
## 2. CI/CD คืออะไร มีความสัมพันธ์กับ DevOps อย่างไร

CI (Continuous Integration) คือ กระบวนการรวม source code ของคนในทีมพัฒนาเข้าด้วยกัน และมีการทดสอบ ด้วย test script เพื่อให้แน่ใจว่าไม่มี ข้อผิดพลาด ในส่วนใดๆ ของโปรแกรม แล้วจึงทำการ commit ไปที่ branch master อีกชั้น.

โดยในการพัฒนานั้น มักใช้ Build Server มาช่วย กล่าวคือจะเริ่มทำการ Integration กันตั้งแต่เมื่อมีการเปลี่ยนแปลง Source Code ที่ Repository กลาง ระบบจะทำการตรวจสอบ Code หลังจากการเปลี่ยนแปลงว่าทำงานร่วมกันได้หรือไม่ตั้งแต่ Compile, Testing

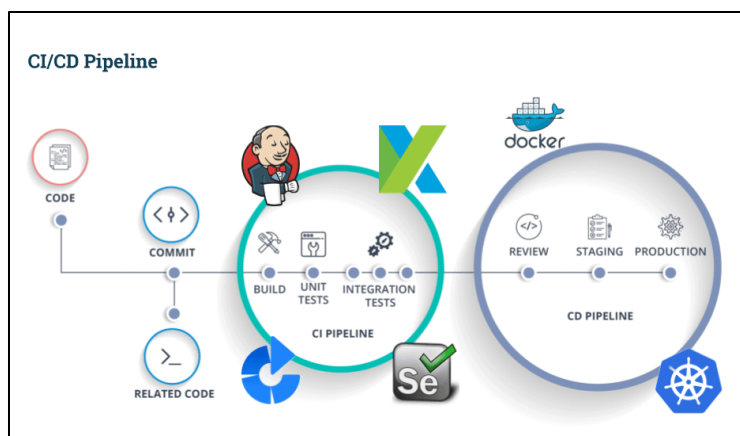
CD (Continuous Deployment) คือ การ Deploy ขึ้น production โดยจะทำทุกขั้นตอน ตั้งแต่ complied build ไปจนถึง deploy ขึ้น production แบบอัตโนมัติทั้งหมด.

CD (Continuous Delivery) คือ การทำทุกขั้นตอนคล้ายกับกับ Continuous Deployment ต่างกันตรงที่จะไม่มีการ deploy ขึ้น production ขึ้นทันที แต่จะเป็นการทำ manual deploy หรือจะเป็นแบบ one click deploy ก็ได้ หลังจาก QA หรือ ลูกค้า พอใจในตัวผลิตภัณฑ์ที่ทีมทำออกมา.



ที่มา : <https://medium.com/@narissara.tan/การนำ-automated-testing-มารวมเข้ากับ-กระบวนการ-ci-cd-58c97e31e230>

เพื่อแก้ปัญหาในการ deploy code จึงเกิดเป็นแนวทาง CI/CD โดยทำงานตั้งแต่การ Plan, Code, Build, Test, Release, Deploy, Operate, Monitor หรือบางทีเรียกสั้นๆ ว่า Pipeline



ที่มา: <https://www.softmelt.com/article.php?id=664>

1. Developer เมื่อทำการพัฒนา feature เสร็จ จะทำการ build, test และ run บนเครื่องของตัวเอง (Local) เพื่อให้แน่ใจว่าระบบทำงานได้ถูกต้องและให้แน่ใจว่าสิ่งที่เปลี่ยนแปลงไม่กระทบส่วนอื่น ๆ.

2. ทำการดึง source code ล่าสุดจาก Repository ของระบบ เพื่อตรวจสอบว่ามีการเปลี่ยนแปลงหรือไม่ ถ้ามีการเปลี่ยนแปลงก็ให้ทำการรวมหรือ merge ที่เครื่องของ Developer ก่อน จากนั้นจึงทำการ build, test และ run อีกรอบ เมื่อทุกอย่างผ่านทั้งหมด ให้ทำการส่งการเปลี่ยนแปลงไปยัง Repository กลาง.

3. เมื่อ Repository กลางมีการเปลี่ยนแปลง จะต้องมียระบบ 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 ของทดสอบ ซึ่งเป็นชุดทดสอบย่อย ๆ ไม่เยอะมากเฉพาะในส่วน of 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 ซ้ำ ๆ เพื่อ fixed bug จากที่ QA เจอ หรือที่พบเจอจากการ run regression test แล้ว fail ก็จะเป็นการวน loop ตั้งแต่ต้นจนจบ จนกระทั่งทุกอย่างผ่านหมด Business ฟันธง มาว่าเอาขึ้น production ได้ เป็นการ confirm ว่าเราจะเอา code version สุดท้ายนี้ ขึ้นไปที่ production environment

### 3. หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์จะต้องทำอย่างไรบ้าง

- 1 เพิ่มความเร็วในการทำ CI/CD Pipeline และการเตรียม Environment สำหรับการพัฒนา Software
- 2 วางระบบ Kubernetes ที่มีประสิทธิภาพสูง พร้อมสำหรับการพัฒนา Cloud-Native Application ภายในองค์กร
- 3 Build Software หลายโครงการพร้อมได้ด้วยความเร็วสูง ไม่มีคอขวดด้านระบบ Storage อีกต่อไป
- 4 ทำการ Test ได้อย่างรวดเร็ว สามารถนำข้อมูลจากระบบ Production มาทำการทดสอบและวิเคราะห์แก้ไขปัญหาได้ทันทีที่ต้องการ
- 5 ใช้แนวทาง Infrastructure-as-Code ได้อย่างเต็มที่ รองรับการทำ Automation ครบวงจร

## อ้างอิง

[1] SoftMelt.com. (2563). การพัฒนาระบบตามแนวทาง CI/CD และ DevOps คืออะไร?.

สืบค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก <https://www.softmelt.com/article.php?id=664>

[2] Pariwat Saknimitwong. (2017). Learn DevOps ตอนที่ 2 : DevOps คืออะไร ?.

สืบค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก [https://medium.com/@pariwat\\_s/learn-devops-ตอนที่-2-devops-คืออะไร-18ac48d73625](https://medium.com/@pariwat_s/learn-devops-ตอนที่-2-devops-คืออะไร-18ac48d73625)

[3] narissara. (2017). การนำ Automated testing มารวมเข้ากับ กระบวนการ CI/CD.

สืบค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก <https://medium.com/@narissara.tan/การนำ-automated-testing-มารวมเข้ากับ-กระบวนการ-ci-cd-58c97e31e230>