

DevOps คืออะไร

เป็นการรวมแนวการทำงาน แนวความคิด และเครื่องมือ ที่ช่วยให้การทำงานภายในองค์กรหรือทีมมีประสิทธิภาพมากขึ้น หรือใช้เวลาในการทำงานให้เสร็จ 1 งานนั้นรวดเร็ว โดยเป็นการพัฒนาและแก้ไขโดยทันที และมีความรวดเร็วมากกว่าการพัฒนาหรือสร้างซอฟต์แวร์และการจัดการทำงานของคนในองค์กร โดยการทำงานแบบนี้จะสามารถให้คนภายในองค์กรได้สื่อสารกับลูกค้าและเข้าใจความต้องการของลูกค้าได้ รวมไปถึงงานที่ออกมานั้นอาจจะสมบูรณ์โดยกินเวลาไม่นานในการพัฒนาและเหลือเพียงการปรับปรุงเท่านั้น

หลักการทำงานของ DevOps เป็นการทำงานแบบเป็นทีมโดยการรวมทั้งฝั่งทีมพัฒนาและทีมปฏิบัติเข้าด้วยกัน ตลอดการทำงานของโปรแกรมหนึ่ง โดยทุกคนที่อยู่งานเดียวกันนั้นจะเข้าใจหลักการเบื้องต้น และเหตุผลในการพัฒนาโปรแกรมตัวนั้นๆ โดยในการทำงานนั้นอาจจะมีการนำเครื่องมือสมัยใหม่เข้ามาช่วยจัดการตารางงาน การที่จะช่วยให้เกิดการทำงานที่รวดเร็วขึ้น และเมื่อเกิดปัญหาคนจากทั้งสองทีมก็จะรับทราบและสามารถแก้ไขได้ทันที[1] [2]

ประโยชน์ของ DevOps

1. **การทำงานอย่างรวดเร็ว** เนื่องจากกระบวนการทำงานสามารถเริ่มงานได้ทุกส่วนทันทีจึงทำให้งานที่ทำนั้นมีความรวดเร็ว
2. **ความน่าเชื่อถือของงาน** โปรแกรมหรือซอฟต์แวร์ที่พัฒนาขึ้นมานั้นได้ถูกตรวจสอบจากทั้งสองทีมตลอดเวลาจึงทำให้มีความถูกต้องและไม่ผิดพลาดประสงค์ของงาน
3. **ขนาดของงาน** จากกระบวนการทำงานจะเห็นได้ว่า การทำงานแบบ DevOps จะช่วยในการลดขนาดของงานลงไม่ว่าเป็นโครงสร้างพื้นฐานเป็นโค้ด หรือการใช้เครื่องมืออื่นๆมาช่วยทำงาน
4. **การทำงานร่วมกันที่ปรับปรุงให้ดีขึ้น** เป็นการทำงานของทั้งสองฝ่ายโดยที่ไม่ต้องมีฝ่ายใดรับผิดชอบงานใหญ่เพียงฝ่ายเดียวแต่ทุกๆฝ่ายก็จะช่วยจัดการงานซึ่งกันและกันและทำให้มีความสัมพันธ์ที่ดีของทีม
5. **ความปลอดภัย** ในการทำงานมีการควบคุมที่ละเอียดและยังมีการกำหนดวันเวลาซึ่ง ทุกๆคนภายในทีมจะต้องอยู่ภายใต้กรอบเวลานี้หรือนโยบายต่างๆที่ทางทีมได้ทำการตกลงกันไว้แล้ว

DevOps มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร

ในปัจจุบันซอฟต์แวร์และอินเทอร์เน็ตได้มามีบทบาทในทุกๆอุตสาหกรรมหรือชีวิตประจำวัน ซึ่งทำให้ซอฟต์แวร์ไม่ได้สร้างขึ้นเพื่อธุรกิจ แต่ได้ถูกพัฒนาขึ้นเพื่อเป็นส่วนประกอบอื่นๆ ไม่ว่าจะเป็นซอฟต์แวร์ที่ช่วยเหลือผู้พิการ ซอฟต์แวร์ช่วยในการค้นหาเส้นทาง และอื่นๆ ซึ่งเห็นได้ว่าการที่จะสร้างซอฟต์แวร์ขึ้นมานั้น หากสามารถเข้าถึงอุปกรณ์บนอินเทอร์เน็ตได้ทุกชนิดหรืออุปกรณ์อื่นๆ หลักการทำงานในการสร้างซอฟต์แวร์จึงมีความสำคัญมากๆ เพราะหลักการในการทำงานนั้น จะส่งผลให้เห็นได้ว่างานที่ทำออกมานั้นมีประสิทธิภาพ และใช้เวลาได้คุ้มค่าหรือไม่

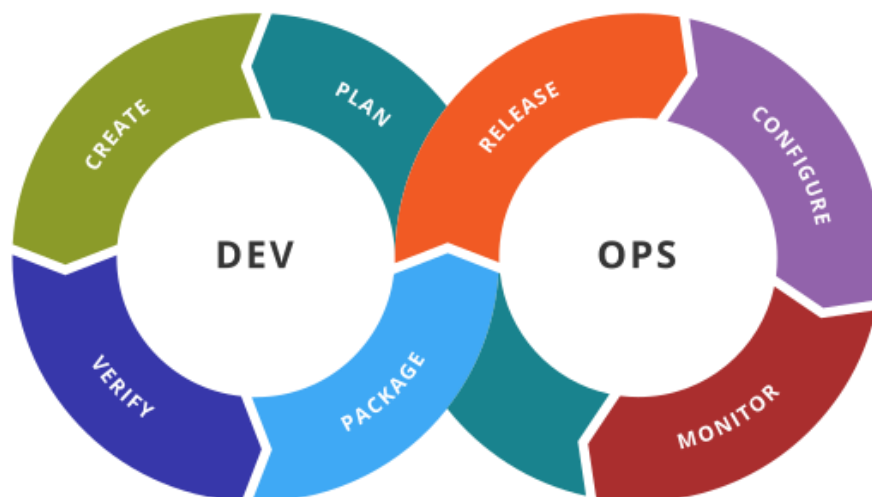
CI/CD คืออะไร

CI (Continuous Integration) คือกระบวนการรวมภาษาคอมพิวเตอร์ที่ผู้พัฒนาทุกคน เพื่อทดสอบว่าสามารถใช้งานได้จริงโดยที่ไม่มีข้อผิดพลาดใดๆ

CD (Continuous Deployment) คือการปล่อยซอฟต์แวร์เข้าสู่ระบบโดยเป็นการทำงานตั้งแต่ต้นของซอฟต์แวร์ไปจนถึงการเปิดใช้งาน โดยการทำงานทั้งหมดเป็นแบบอัตโนมัติ

CD (Continuous Delivery) เหมือนกับการทำงานของ Continuous Deployment แต่มีข้อแตกต่างคือจะเป็นการทำงานแบบไม่อัตโนมัติ คือผู้พัฒนาหรือผู้เปิดใช้งานต้องทำการออกคำสั่งด้วยตนเอง

เป็นกระบวนการทำงานหรือ Pipeline ในการทำงานแบบต่อเนื่องซึ่งหลักจากผ่านการพัฒนาแล้วก็จะทำการเปิดให้ใช้งานและวนเป็นรูปแบบต่อเนื่องไปเรื่อยๆจนกว่าซอฟต์แวร์ตัวนั้นจะหยุดพัฒนา[3] [4]



ภาพจาก <https://www.redmineup.com/pages/blog/devops-in-redmine>

1. **วางแผน** เป็นการเริ่มต้นโดยทุกฝ่ายที่เกี่ยวข้องจะมีการวางแผนกันอย่างรอบคอบ เช่น พัฒนาด้วยภาษาใด ใช้บน Platform ใด ฯลฯ ซึ่งเป็นการวางแผนงานทั้งหมด รวมทั้ง ข้อจำกัดของงานและขอบเขตของงานซึ่งคล้ายคลึงวิธีการทางวิทยาศาสตร์
2. **สร้าง** เป็นขั้นตอนที่เขียนโปรแกรมหรือตั้งค่าให้ผู้ที่เขียนโปรแกรมเมอร์ทุกคนสามารถ ทำงานได้ภายใน Environment เดียวกัน เพื่อให้สามารถทำงานได้โดยไม่ต้องรอกันและ สามารถที่จะนำโปรแกรมของใครในทีมไปทำงานในเครื่องใดก็ได้
3. **การตรวจสอบ** เป็นการตรวจสอบว่าโปรแกรมที่ทำขึ้นมา มีข้อผิดพลาดหรือมีการเพิ่ม คุณสมบัติบางส่วนที่ทำให้วัตถุประสงค์ที่วางแผนไว้คลาดเคลื่อนหรือไม่ หากมีข้อผิดพลาด ก็ให้ทำการแก้ไขโดยทันที
4. **แพ็คเกจ** คือการส่งโปรแกรมที่ทำขึ้นมานั้นให้เข้าสู่ระบบ Server เพื่อให้เกิดความง่ายและ สะดวกในการตรวจสอบหรือเป็นการทดสอบเบื้องต้นว่าโปรแกรมหากว่านั้นสามารถ ทำงานได้อย่างถูกต้องหรือไม่
5. **เปิดให้ใช้งาน** เมื่อทำการส่งโปรแกรมเข้าสู่ Server แล้วก็จะมีการเปิดใช้งานไปยัง Deployment Platform ที่เป็นไปตามที่วางแผนเอาไว้
6. **การตั้งค่า** โปรแกรมนั้นมีความจำเป็นที่จะต้องได้รับการตั้งค่า เพื่อให้การทำงานของ โปรแกรมไม่มีข้อผิดพลาดหรือการแยก โปรแกรมที่ใช้งานในบริษัท กับโปรแกรมที่ใช้งาน สำหรับผู้พัฒนา เพื่อให้เมื่อเกิดการแก้ไขจะไม่ต้องไปแก้ไขภายในโปรแกรมที่ได้เปิดใช้งาน ไปแล้ว
7. **การควบคุม** เป็นการตรวจสอบขั้นสุดท้ายว่าโปรแกรมสามารถทำงานได้อย่างปกติไม่มี ข้อบกพร่องใดๆ ซึ่งหากเกิดข้อผิดพลาดขึ้นก็ให้ทำการ เริ่มจากการวางแผนใหม่

CI/CD มีความสัมพันธ์กับ DevOps อย่างไร

ในการทำงานหรือการพัฒนาซอฟต์แวร์ขึ้นมานั้น งานจะไม่จบลงทันทีที่จะมีการทำงานต่อไปจนกว่า ซอฟต์แวร์ดังกล่าวหยุดพัฒนา ซึ่งหลักการของ CI/CD นั้นได้มีการกล่าวถึงการจัดการตั้งแต่ ขั้นตอนพัฒนาไป จนถึงขั้นตอนการเปิดให้ใช้งาน จะเห็นได้ว่าต่อให้มีการเปิดใช้งานแล้วก็ยังคงต้องมีการพัฒนาต่อไป ซึ่งหลักการ ของ CI/CD จะช่วยให้การทำงานแบบ DevOps มีประสิทธิภาพมากขึ้นเนื่องจากจะลดระยะเวลาในการพัฒนา ลง ลดต้นทุน เมื่อเกิดการแก้ไขหรือข้อผิดพลาดก็จะสามารถแก้ไขได้อย่างทันที ซอฟต์แวร์นั้นจะมีความยืดหยุ่น ที่สูงอีกด้วย

หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์ จะต้องทำอย่างไรบ้าง

เริ่มจากการทำให้ทุกคนภายในทีมหรือองค์กรนั้นๆ ทำการศึกษาและค้นคว้า เพื่อให้เข้าใจความหมาย หลักการ วิธีการทำงาน ของ DevOps และ CI/CD ก่อน และค่อยเริ่มลงมือทำโดยในการลงมือทำนั้นในช่วงแรกๆจะต้องไปอย่างช้าๆเพื่อให้คนในทีมมีความเข้าใจที่ตรงกันและไม่ให้เกิดข้อผิดพลาดของคนในทีม เมื่อในการทำงานเริ่มที่จะไปได้ดีก็ค่อยๆทำการปรับเปลี่ยนการทำงานให้เข้าใกล้หลักการของ DevOps และ CI/CD และทำไปเรื่อยๆ จนเกิดความเคยชินในการทำงานด้วยหลักการทั้งสองแบบ

อ้างอิง

- [1] DevOps คืออะไร - Amazon Web Services (AWS). Amazon Web Services, Inc n.d.
<https://aws.amazon.com/th/devops/what-is-devops/> (accessed February 1, 2021).
- [2] DevOps CI/CD คืออะไร? – Ton’s Blog n.d. <https://ton.packetlove.com/blog/life-style/devops-ci-cd.html> (accessed February 1, 2021).
- [3] Junthep T. เมื่อหัดทำ CI/CD ครั้งแรก. Medium 2018. <https://medium.com/@thraithep.a/เมื่อหัดทำ-ci-cd-ครั้งแรก-f4f79151311c> (accessed February 1, 2021).
- [4] Administrator. การพัฒนาระบบตามแนวทาง CI/CD และ DevOps คืออะไร? n.d.
<https://www.softmelt.com/article.php?id=664> (accessed February 1, 2021).