

DevOps

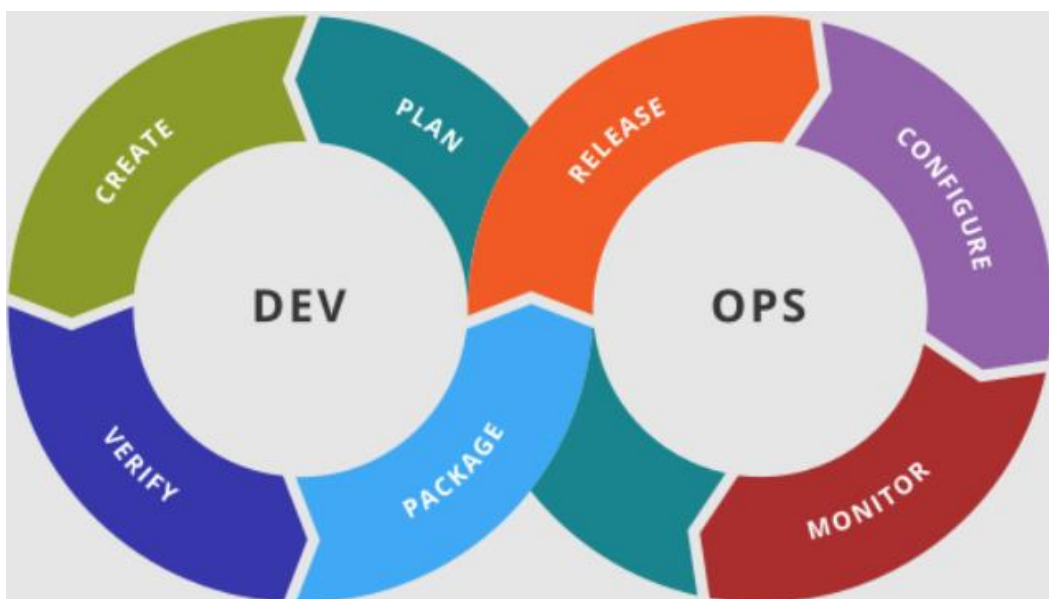
นิยามโมเดล DevOps

คือการผสมผสานแนวความคิดเชิงวัฒนธรรม แนวทางปฏิบัติ และเครื่องมือต่างๆ ที่ช่วยเพิ่มความสามารถขององค์กรในการส่งมอบแอปพลิเคชันและบริการอย่างรวดเร็ว โดยพัฒนาและปรับปรุงผลิตภัณฑ์ต่างๆ ให้เร็วกว่ากระบวนการการพัฒนาซอฟต์แวร์และการจัดการโครงสร้างพื้นฐานแบบดั้งเดิม ความรวดเร็วนี้ช่วยให้องค์กรสามารถให้บริการแก่ลูกค้าของตนได้ดีขึ้น และสามารถแข่งขันในตลาดได้อย่างมีประสิทธิภาพมากขึ้น[1]

หลักการทำงาน

ในการทำงานแบบ DevOps ทีม Development และทีม Operations จะทำงานร่วมกันเป็นทีมเดียวกัน ตั้งแต่กระบวนการพัฒนา การทดสอบไปจนถึงการปรับใช้และการปฏิบัติงาน เปลี่ยนกระบวนการต่างๆ ที่เคยทำงานแบบ Manual ให้เป็น Automation มากขึ้นเพื่อความรวดเร็ว ใช้เทคโนโลยีและเครื่องมือต่างๆ เพื่อช่วยให้พัฒนา Software ได้อย่างรวดเร็วและมีคุณภาพที่เชื่อถือได้[3]

DevOps จะทำให้เกิดการผลักดันให้ทีม Development มีส่วนร่วมในการองค์กรมากขึ้น นอกเหนือจากบทบาทที่เกี่ยวข้องในการพัฒนา เช่น การดำเนินงาน infrastructure และ monitoring, การสนับสนุนลูกค้า และการวิเคราะห์ การเปลี่ยนแปลงนี้จะขยายเครือข่ายของคนทำงานร่วมกัน ซึ่งร่วมไปถึงการขยายขอบเขตและลักษณะของการทดสอบ ซึ่งในภาพที่ 1 จะเป็นกระบวนการทำงานแบบ DevOps



ภาพที่ 1 กระบวนการทำงานแบบ DevOps [3]

- **Plan** จะเป็นการวางแผนภาพรวมทั้งหมดของงาน โดยต้องช่วยกันทุกฝ่าย เช่น จะพัฒนาด้วยภาษาอะไร Framework อะไร จะรันบน Platform อะไร จะไปลงไว้ที่ไหน จะเก็บข้อมูลอย่างไร จะแก้ปัญหาที่เกิดขึ้น และจะพัฒนา Process การทำงานอย่างไร
- **Create** จะเป็นการลงมือพัฒนาโดยทุกคนจะต้องทำงานบน Environment เดียวกัน
- **Verify** เป็นการตรวจสอบว่าโปรแกรมที่พัฒนาขึ้นมามีปัญหาอะไร หรือสามารถเพิ่มฟีเจอร์อะไรเข้าไปได้บ้าง
- **Package** การจะส่งโปรแกรมขึ้นไปรันบน Server เพื่อความสะดวกรวดเร็ว จะต้องมีการเอาโปรแกรมไปใส่ใน Technology ที่ออกแบบมาเพื่อนำโปรแกรมไปรันได้อย่างราบรื่น
- **Release** การนำโปรแกรมไปรันบน Deployment Platform ที่ต้องการ ตอนนี้กำลังทำของ Development หรือทำหรือ Production อยู่ จะต้องเลือกไป Deploy ให้ถูกที่
- **Configure** เป็นการตั้งค่าโปรแกรม เพื่อให้การทำงานขงโปรแกรมไม่มีข้อผิดพลาด เพื่อไม่ให้เกิดปัญหาหลังจากเปิดใช้งานตัวโปรแกรม แล้วต้องไปแก้ทีหลัง
- **Monitor** การตรวจสอบว่าโปรแกรมมันขึ้นไปแล้วทำงานได้ปกติ มีสุขภาพแข็งแรงดี การดู Log ว่าโปรแกรมเราทำงานมีปัญหาตรงไหน มีความสำคัญอย่างมากในการแก้ไขบั๊ก

ประโยชน์ของ DevOps

1. **ความรวดเร็ว** เนื่องจากการทำงานที่ร่วมหลายทีมเข้าด้วยกัน ทำให้ลดปัญหาที่อาจเกิดภายหลังโดยสามารถส่งงานให้ลูกค้าตลอดเวลา และมีการพัฒนาอย่างต่อเนื่อง
2. **ส่งมอบอย่างรวดเร็ว** จากรูปแบบของ DevOps จะเป็นการทำงานที่มีเป็นการทำงานเริ่มได้ทุกส่วนไม่ได้ต้องรอขั้นตอนเหมือนแบบเดิม สามารถทำให้เราส่งงานลูกค้าได้ตลอดเวลาเพื่อสามารถปรับเปลี่ยน ให้เข้ากับเหตุการณ์ปัจจุบันได้
3. **ความน่าเชื่อถือ** เนื่องจากการส่งงานให้ลูกค้าตลอดเวลาทำให้สามารถมั่นใจได้ว่าตัวซอฟต์แวร์ที่สร้างขึ้นมามีความน่าเชื่อถือ และตรงตามวัตถุประสงค์
4. **ขนาด** การทำงานแบบ DevOps สามารถช่วยให้ขนาดของงานลดลงและ ทำให้งานมีประสิทธิภาพเพิ่มมากยิ่งขึ้น ไม่ว่าจะเป็นการลดโครงสร้างพื้นฐาน
5. **การทำงานร่วมกันที่ปรับปรุงดีขึ้น** โดยการทำงานแบบ DevOps จะทำงานอย่างใกล้ชิดกับลูกค้า ทำให้ทีมพัฒนาสามารถปรับปรุงงานให้ดีขึ้นตามที่ลูกค้าต้องการได้

6. **การรักษาความปลอดภัย** ดำเนินการอย่างรวดเร็วในขณะที่ยังคงรักษาการควบคุมและการปฏิบัติสอดคล้องกับกฎระเบียบ คุณสามารถรับเอาโมเดล DevOps มาใช้โดยไม่ต้องสูญเสียความปลอดภัยโดยใช้นโยบายการปฏิบัติตามกฎระเบียบที่ทำงานอัตโนมัติ การควบคุมโดยละเอียด และเทคนิคการจัดการการกำหนดค่า

DevOps มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร

ในปัจจุบันซอฟต์แวร์และอินเทอร์เน็ตได้เปลี่ยนโลกและอุตสาหกรรมต่าง ๆ มากมาย ตั้งแต่การซื้ออุปโภคบริโภคและความบันเทิงไปจนถึงการธนาคาร ซอฟต์แวร์ไม่ได้เพียงแค่สนับสนุนธุรกิจเท่านั้น แต่ยังเป็นองค์ประกอบสำคัญในทุกภาคส่วนของธุรกิจ บริษัทต่างๆ มีปฏิสัมพันธ์กับลูกค้าของตนผ่านทางซอฟต์แวร์ที่ให้บริการหรือมีแอปพลิเคชันทางออนไลน์บนอุปกรณ์ทุกชนิด นอกจากนั้นยังใช้ซอฟต์แวร์เพื่อเพิ่มประสิทธิภาพการทำงานโดยเปลี่ยนแปลงทุกส่วนของห่วงโซ่คุณค่า เช่น โลจิสติกส์ การสื่อสาร และการปฏิบัติการ ในทำนองเดียวกันกับที่บริษัทขายสินค้าที่จับต้องได้เปลี่ยนวิธีการออกแบบ สร้าง และส่งมอบผลิตภัณฑ์โดยใช้การทำงานอัตโนมัติทางอุตสาหกรรมตลอด[1] ทำให้การพัฒนาซอฟต์แวร์แต่ละตัวนั้นมีความสำคัญเป็นอย่างมาก ทำให้หลักการที่ใช้ในการพัฒนานั้นมีความสำคัญเป็นอย่างมาก เพราะหลักการทำงาน จะทำให้งานที่สร้างนั้นมีประสิทธิภาพเพียงพอ ต่อการใช้งาน มีงบประมาณ และใช้เวลาคุ้มค่าหรือไม่

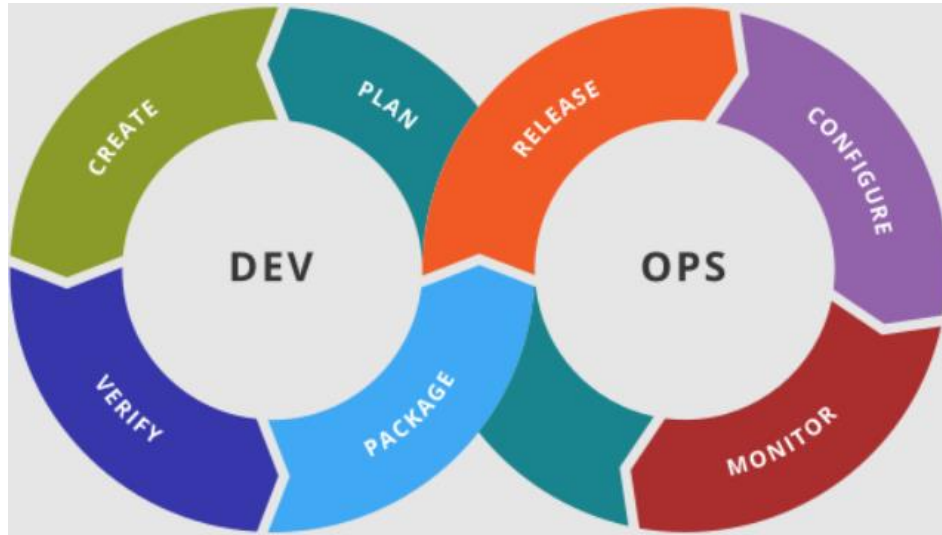
CI/CD

CI/CD คืออะไร

CI ย่อมาจาก Continuous Integration ครับ ซึ่งจะมุ่งเน้นไปที่การที่ Developer สามารถที่จะรวมงานที่ตัวเองทำเข้ากับส่วนกลางได้สะดวกรวดเร็วและสามารถตรวจสอบความถูกต้องได้ด้วยการทำ Automate Testing[4]

CD ย่อมาจาก Continuous Delivery และ Continuous Deployment ครับ ซึ่งจะมุ่งเน้นไปที่ Process ของการนำสิ่งที่พัฒนาใหม่ให้ถึงมือของผู้ใช้ได้เร็วที่สุด เช่นการทำ Automate Deployment ที่ช่วยลดระยะเวลาการติดตั้งงานให้กับลูกค้า รวมถึงเรายังพยายามที่จะส่งมอบงานให้กับผู้ใช้ให้เร็วและส่งมอบอยู่เป็นประจำ เพื่อให้ผู้ใช้ได้มอบ Feedback ต่างๆและปรับแต่ง Application ได้อย่างสม่ำเสมอ[4]

ซึ่งจะเป็นกระบวนการในการทำงาน หรือจะเรียกว่า Pipeline โดยหลักการทำงานจะเป็นการทำงานแบบต่อเนื่อง ตั้งแต่การวางแผนไปจนถึงการส่งมอบงาน แล้วก็จะมีการวนลูปเข้าไปเรื่อย ๆ จนกว่าการพัฒนาซอฟต์แวร์เสร็จสิ้น ซึ่งสามารถดูได้จากภาพที่ 1(ต่อ) ที่เคยได้อธิบายในด้านบนไปแล้ว[5]



ภาพที่ 1(ต่อ) กระบวนการทำงานแบบ DevOps [3]

CI/CD มีความสัมพันธ์กับ DevOps อย่างไร

CI/CD เป็นกระบวนการในการทำงาน(Pipeline) และ เครื่องมือสำคัญที่ใช้ในการพัฒนาซอฟต์แวร์ และถือเป็นส่วนหนึ่งของ DevOps ที่ช่วยเพิ่ม fast & direct feedback ให้กับทุกฝ่าย[6]

หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์จะต้องทำอย่างไรบ้าง

- ให้ทุกคนศึกษาเกี่ยวกับ DevOps และ CI/CD ว่าคืออะไร ทำงานอย่างไร
- ลงมือปฏิบัติทีละขั้นตอนตามที่เข้าใจ ซึ่งในช่วงแรกการทำงานอาจจะยังไม่มีประสิทธิภาพ อาจจะต้องมีการปรับตัว
- มีการพูดคุยกันว่าเกิดปัญหาอะไรบ้างไม่เข้าใจตรงไหน แล้วก็ค่อย ๆ อธิบาย และปรับกระบวนการไปเรื่อย ๆ จนกว่าจะเป็นการนำ DevOps และ CI/CD จริงๆ
- ซึ่งหากทำสำเร็จจะเป็นการพัฒนาองค์กรให้มีการพัฒนาซอฟต์แวร์ที่มีประสิทธิภาพมากยิ่งขึ้น

อ้างอิง

[1] Amazon Web Services.(ม.ป.ป). **DevOps คืออะไร**. ค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก <https://aws.amazon.com/th/devops/what-is-devops/#:~:text=DevOps%20คือการผสมผสานแนว,บริการแก่ลูกค้าของคุณ>

[2] apiwat sangakong.(2020). **What is DevOps?**. Retrieved February 25 2021, from <https://welovebug.com/what-is-devops-e9596f144738>

[3] SPICYDOG.(2562). **DevOps คืออะไร นำมาประโยชน์ได้ยังไง และตัวอย่างการทำ DevOps ที่ Credit OK**. ค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก <https://www.spicydog.org/blog/introduction-to-devops-and-the-practical-use-cases-at-credit-ok/>

[4] Ponggun.(2562).**ติดตั้ง Jenkins บน Windows Server 2012 เพื่อทำ CI/CD ASP.Net 4.5 ร่วมกับ GitLab และ IIS**. ค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก <https://ponggun.medium.com/ติดตั้ง-jenkins-บน-windows-server-2012-เพื่อทำ-ci-cd-asp-net-4-5-ร่วมกับ-gitlab-และ-iis-47c1ad249b70>

[5] Ton's Blog.(2562). **DevOps CI/CD คืออะไร?**. ค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก [https://ton.packetlove.com/blog/life-style/devops-ci-cd.html#:~:text=CI%2FCD%20\(Continuous%20Integration%2C,น่าจะเป็น%20Jenkins%20ที่](https://ton.packetlove.com/blog/life-style/devops-ci-cd.html#:~:text=CI%2FCD%20(Continuous%20Integration%2C,น่าจะเป็น%20Jenkins%20ที่)

[6] Tanin Asi.(2562). **CI/CD : Why and How**. ค้นเมื่อ 25 กุมภาพันธ์ 2564, จาก <https://medium.com/@taninasi/ci-cd-why-and-how-99d94ffaa147>