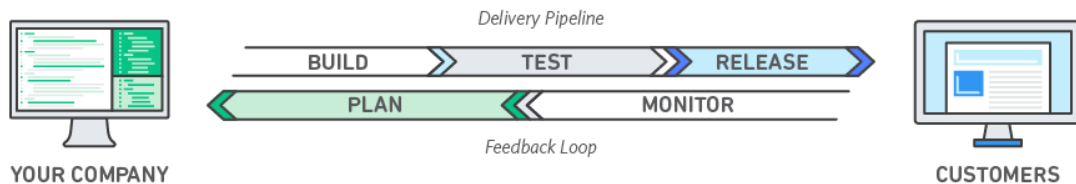


นิยามโมเดล DevOps

DevOps คือการผสมผสานแนวความคิดเชิงวัฒนธรรม แนวทางปฏิบัติ และเครื่องมือต่างๆ ที่ช่วยเพิ่มความสามารถขององค์กรในการส่งมอบแอปพลิเคชันและบริการอย่างรวดเร็ว โดยพัฒนาและปรับปรุงผลิตภัณฑ์ต่างๆ ให้เร็วกว่ากระบวนการการพัฒนาซอฟต์แวร์และการจัดการโครงสร้างพื้นฐานแบบดั้งเดิม ความรวดเร็วนี้ช่วยให้องค์กรสามารถให้บริการแก่ลูกค้าของตนได้ดีขึ้น และสามารถแข่งขันในตลาดได้อย่างมีประสิทธิภาพมากขึ้น



ภาพที่ 1 ภาพอธิบายการทำงานของโมเดล DevOps

1.DevOps ทำงานอย่างไร

สำหรับโมเดล DevOps ทีมพัฒนาและทีมปฏิบัติการจะไม่ทำงานแบบ “ต่างคนต่างทำ” อีกต่อไป บางครั้ง ทั้งสองทีมจะจับมือร่วมงานเป็นทีมเดียวกันโดยที่เหล่าวิศวกรจะทำงานตลอดวงจรการทำงานของแอปพลิเคชัน ตั้งแต่การพัฒนาและการทดสอบไปจนถึงการปรับใช้และการปฏิบัติงาน และพัฒนาขอบเขตความสามารถที่ไม่ได้จำกัดอยู่ที่การทำงานเพียงอย่างเดียว

ในบางโมเดลของ DevOps ทีมประกันคุณภาพและทีมรักษาความปลอดภัยอาจทำงานร่วมกับทีมพัฒนาและทีมปฏิบัติการอย่างใกล้ชิดยิ่งขึ้นตลอดวงจรการทำงานของแอปพลิเคชัน ในเวลาที่มีการรักษาความปลอดภัยเป็นจุดสำคัญของทุกคนในทีม DevOps เราจะเรียกว่า DevSecOps

ทีมต่างๆ ใช้ข้อปฏิบัติในการเปลี่ยนกระบวนการต่างๆ ที่เคยทำงานแบบแมนนวลและเชิงซ้ำให้ทำงานอัตโนมัติ พวกเขาใช้ชุดเทคโนโลยีและเครื่องมือต่างๆ ที่ช่วยให้พวกเขาดำเนินการและพัฒนาแอปพลิเคชันได้อย่างรวดเร็วและเชื่อถือได้ นอกจากนั้น เครื่องมือเหล่านี้ยังช่วยให้วิศวกรเหล่านี้ทำงานได้อย่างอิสระ (เช่น การปรับใช้โค้ด หรือการจัดเตรียมโครงสร้างพื้นฐาน) ที่โดยปกติแล้วจำเป็นต้องได้รับความช่วยเหลือจากทีมอื่น พร้อมทั้งยังช่วยทีมงานทำงานได้รวดเร็วยิ่งขึ้นอีกด้วย

2.ประโยชน์ของ DevOps

2.1 ความรวดเร็ว

ดำเนินการอย่างรวดเร็วเพื่อให้คุณสามารถสร้างสรรค์นวัตกรรมให้แก่ลูกค้าได้รวดเร็วยิ่งขึ้น ปรับตัวต่อการเปลี่ยนแปลงของตลาดได้ดียิ่งขึ้น และมีประสิทธิภาพยิ่งขึ้นในการกระตุ้นให้เกิดผลลัพธ์ทางธุรกิจ โมเดล DevOps จะทำให้นักพัฒนาและทีมปฏิบัติการของคุณบรรลุผลดังกล่าวได้ ตัวอย่างเช่น ไมโครเซอร์วิสและการส่งมอบอย่างต่อเนื่องจะทำให้ทีมมีความเป็นเจ้าของบริการและออกอัปเดตให้แก่ลูกค้าได้รวดเร็วยิ่งขึ้น

2.2 การส่งมอบอย่างรวดเร็ว

ออกรุ่นใหม่ๆ ถีขึ้นและรวดเร็วยิ่งขึ้น เพื่อให้คุณสามารถสร้างสรรค์นวัตกรรมและปรับปรุงผลิตภัณฑ์ของคุณได้รวดเร็วยิ่งขึ้น ยิ่งคุณสามารถออกคุณสมบัติใหม่ๆ และแก้ไขจุดบกพร่องได้เร็วเท่าไร คุณก็ยิ่งตอบสนองความต้องการของลูกค้าและสร้างความได้เปรียบในการแข่งขันได้เร็วเท่านั้น การบูรณาการอย่างต่อเนื่อง และการส่งมอบอย่างต่อเนื่อง เป็นข้อปฏิบัติที่ทำให้กระบวนการออกซอฟต์แวร์ทำงานอัตโนมัติตั้งแต่การสร้างไปจนถึงการปรับใช้

2.3 ความเชื่อถือได้

ทำให้แน่ใจถึงคุณภาพของการอัปเดตแอปพลิเคชันและการเปลี่ยนแปลงโครงสร้างพื้นฐาน เพื่อให้คุณสามารถส่งมอบได้อย่างน่าเชื่อถือในความเร็วที่มากขึ้น ในขณะที่ยังคงรักษาประสิทธิภาพใช้งานเชิงบวกของผู้ใช้ปลายทาง ใช้ข้อปฏิบัติอย่างเช่น การบูรณาการอย่างต่อเนื่อง และการส่งมอบอย่างต่อเนื่อง เพื่อทดสอบว่าการเปลี่ยนแปลงนั้นทำงานได้และปลอดภัยหรือไม่ ข้อปฏิบัติในการตรวจสอบและการบันทึกจะช่วยให้คุณทราบถึงประสิทธิภาพการทำงานในทันที

2.4 ขนาด

ดำเนินการและจัดการโครงสร้างพื้นฐานและกระบวนการพัฒนาของคุณในขนาดต่างๆ การทำงานอัตโนมัติและความสอดคล้องจะช่วยให้คุณจัดการระบบที่ซับซ้อนหรือมีการเปลี่ยนแปลงได้อย่างมีประสิทธิภาพ โดยมีความเสี่ยงที่ลดลง ตัวอย่างเช่น โครงสร้างพื้นฐานเป็นโค้ดจะช่วยให้คุณในการจัดการสภาพแวดล้อมในการพัฒนา การทดสอบ และการทำงานจริงในลักษณะที่ทำซ้ำได้และมีประสิทธิภาพยิ่งขึ้น

2.5 การทำงานร่วมกันที่ปรับปรุงดีขึ้น

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

2.6 การรักษาความปลอดภัย

ดำเนินการอย่างรวดเร็วในขณะที่ยังคงรักษาการควบคุมและการปฏิบัติตามข้อกำหนดกับกฎระเบียบ คุณสามารถรับเอาโมเดล DevOps มาใช้โดยไม่ต้องสูญเสียความปลอดภัยโดยใช้นโยบายการปฏิบัติตามกฎระเบียบที่ทำงานอัตโนมัติ การควบคุมโดยละเอียด และเทคนิคการจัดการการกำหนดค่า ตัวอย่างเช่น คุณสามารถใช้โครงสร้างพื้นฐานเป็นโค้ดและนโยบายเป็นโค้ด เพื่อกำหนดและติดตามการปฏิบัติตามกฎระเบียบในขนาดต่างๆ

3.ความสำคัญของ DevOps

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

4.แนวปฏิบัติของ DevOps

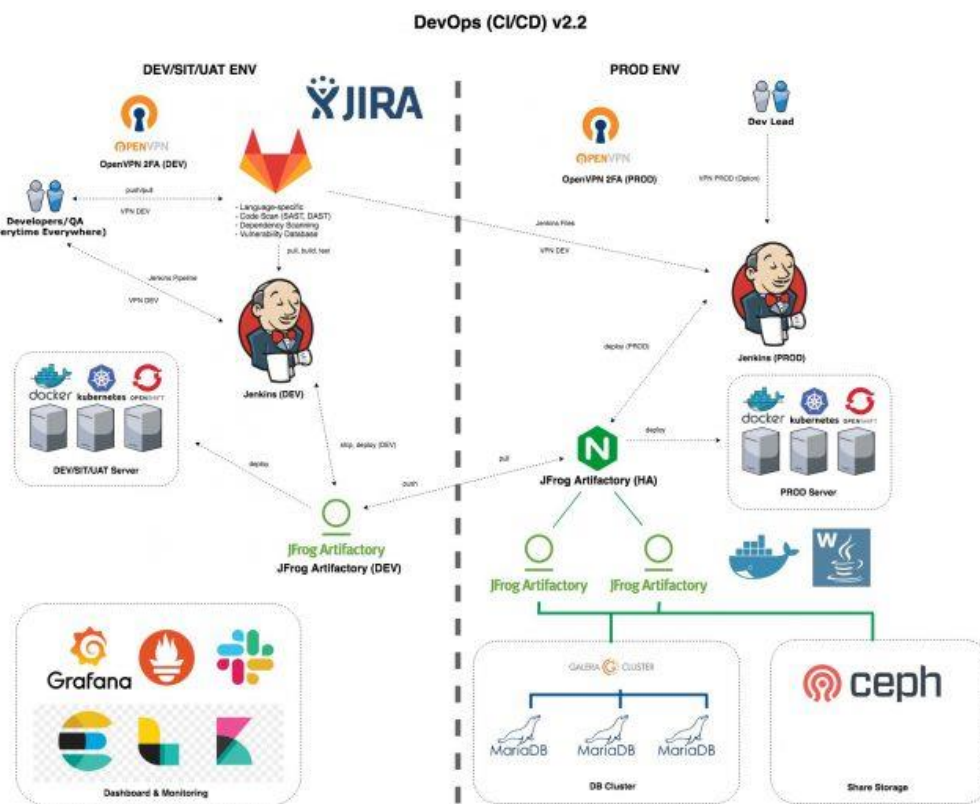
มีข้อปฏิบัติสำคัญไม่กี่ข้อที่จะช่วยให้องค์กรสร้างสรรค์นวัตกรรมได้รวดเร็วยิ่งขึ้น โดยการปรับปรุงขั้นตอนและเปลี่ยนกระบวนการพัฒนาซอฟต์แวร์และการจัดการโครงสร้างพื้นฐานให้ทำงานอัตโนมัติ ข้อปฏิบัติส่วนใหญ่ต้องดำเนินการโดยใช้เครื่องมือที่เหมาะสม

ข้อปฏิบัติพื้นฐานข้อหนึ่งคือ การอัปเดตบ่อยๆ ที่ละเล็กละน้อย วิธีจะทำให้องค์กรสร้างสรรค์นวัตกรรมให้แก่ลูกค้าได้เร็วยิ่งขึ้น โดยทั่วไปแล้ว อัปเดตเหล่านี้มีลักษณะเป็นส่วนเพิ่มมากกว่าอัปเดตเป็นครั้งคราวที่ทำภายใต้ข้อปฏิบัติในการออกอัปเดตแบบเดิมๆ การอัปเดตบ่อยๆ ที่ละเล็กละน้อยจะทำให้การปรับใช้ในแต่ละครั้งมีความเสี่ยงลดลง โดยจะช่วยให้ทีมแก้ไขจุดบกพร่องได้เร็วขึ้น เนื่องจากทีมสามารถระบุการปรับใช้ครั้งสุดท้ายที่ทำให้เกิดปัญหาได้ แม้ว่าจังหวะเวลาและขนาดของอัปเดตจะแตกต่างกันก็ตาม แต่องค์กรที่ใช้โมเดล DevOps จะปรับใช้อัปเดตได้บ่อยกว่าองค์กรที่ใช้ข้อปฏิบัติในการพัฒนาซอฟต์แวร์แบบเดิมๆ มาก

องค์กรยังอาจต้องใช้สถาปัตยกรรมไมโครเซอร์วิสอีกด้วย เพื่อให้แอปพลิเคชันมีความยืดหยุ่นมากขึ้น และสามารถสร้างสรรค์นวัตกรรมได้เร็วขึ้น สถาปัตยกรรมไมโครเซอร์วิสจะแยกระบบขนาดใหญ่ที่มีความซับซ้อนออกเป็นโปรเจกต์ต่างๆ ที่ทำงานอิสระ แอปพลิเคชันจะถูกแบ่งออกเป็นแต่ละส่วนประกอบ (บริการ) จำนวนมาก โดยแต่ละบริการจะถูกกำหนดขอบเขตให้มีจุดประสงค์เดียวหรือการทำงานอย่างเดียว และทำงานโดยอิสระจากทั้งบริการในระดับเดียวกันและแอปพลิเคชันโดยรวม สถาปัตยกรรมดังกล่าวจะลดค่าใช้จ่ายในการประสานงานในการอัปเดตแอปพลิเคชัน และเมื่อแต่ละบริการถูกมอบหมายให้กับทีมขนาดเล็กที่มีความคล่องตัวโดยเป็นมีความเป็นเจ้าของแต่ละบริการแล้ว องค์กรจะสามารถดำเนินการได้รวดเร็วมากขึ้น

อย่างไรก็ตาม การนำไมโครเซอร์วิสมารวมเข้ากับการออกอัปเดตในความถี่ที่เพิ่มขึ้นนั้น อาจนำไปสู่การปรับใช้งานมากขึ้นอย่างมีนัยสำคัญ ซึ่งอาจก่อให้เกิดความท้าทายในการปฏิบัติงานได้ ด้วยเหตุนี้ ข้อปฏิบัติของ DevOps อย่างเช่น การบูรณาการอย่างต่อเนื่องและการส่งมอบอย่างต่อเนื่อง จึงช่วยแก้ไขปัญหาดังกล่าวได้ และช่วยให้องค์กรส่งมอบได้อย่างรวดเร็วในลักษณะที่ปลอดภัยและเชื่อถือได้ ข้อปฏิบัติสำหรับการทำงานอัตโนมัติของโครงสร้างพื้นฐาน เช่น โครงสร้างพื้นฐานเป็นโค้ดและการจัดการการกำหนดค่า จะช่วยให้ทรัพยากรในการประมวลผลมีความยืดหยุ่นและตอบสนองต่อการเปลี่ยนแปลงที่บ่อยครั้ง นอกจากนั้น การใช้งานการตรวจสอบและการบันทึกจะช่วยให้วิศวกรสามารถติดตามประสิทธิภาพของแอปพลิเคชันและโครงสร้างพื้นฐานได้ เพื่อให้สามารถตอบสนองต่อปัญหาได้อย่างรวดเร็ว

DevOps CI/CD



ภาพที่2 ภาพแสดงความหมายของ CI/CD

5.นิยาม CI

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

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

6.นิยาม CD

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

2. CD (Continuous Delivery) คือ การทำทุกขั้นตอนคล้ายกันกับ Continuous Deployment ต่างกันตรงที่จะไม่มีการ deploy ขึ้น production ขึ้นในทันที แต่จะเป็นการทำ manual deploy หรือจะเป็นแบบ one click deploy ก็ได้ หลังจาก QA หรือ ฝ่าย Business พอใจในตัว product ที่ทีมทำออกมา

ขอบเขตของการทำงาน ของ DevOps แต่ละที่ จะไม่เหมือนกัน ขึ้นอยู่กับ scope ที่ทำได้ มีระดับไหน บางที่ ก็คือทำตั้งแต่ต้นน้ำ ยันปลายน้ำ คือตั้งแต่วางแผน สร้าง Infrastructure เอง ทำ ENV ให้ Dev ใช้ เขียน Pipeline ตลอดจน ทำ Load Test, Performance Test, Security Test และระบบ Monitor & Alert เอง ทั้งหมด แบบนี้ก็ดีตรงที่จะรู้และเข้าใจ ในแต่ละส่วนอย่างดี ทำให้งานออกมามีประสิทธิภาพ ควบคุมได้ แต่ถ้ามีหลายๆ Project ก็คงทำแบบนี้ไม่ไหว ..

บางที่ DevOps จะมีหน้าที่แค่ทำระบบให้ Dev มาใช้งาน แต่จะไม่มีสิทธิ ในการทำอย่างอื่นเอง แบบด้านบน แบบนี้ ก็จะทำให้ ควบคุมอะไรไม่ได้ทั้งหมด แต่ถ้ามีหลายๆ Project ก็คงรองรับ

7. ขั้นตอนการพัฒนาระบบตามแนวทาง CI/CD

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 จะต้องมีการ 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 ซึ่งเป็นชุดทดสอบย่อยๆ ไม่เยอะมากเฉพาะในส่วนของ 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 ทุกอย่าง

8. ข้อดีของการพัฒนาระบบตามแนวทาง CI/CD

ลดระยะเวลาในการพัฒนา ลดต้นทุนในการพัฒนา รองรับการเปลี่ยนแปลงแก้ไขได้ง่าย ดูแลรักษาได้ง่าย เนื่องจากการดำเนินงานไปอย่างต่อเนื่อง เป็นไปอย่างอัตโนมัติและทราบผลลัพธ์ได้ทันที ทำให้เราสามารถส่งมอบแอปฯ รุ่นใหม่ให้กับลูกค้าได้ไวขึ้น