

DevOps คืออะไร

DevOps คือการผสมผสานแนวความคิดเชิงวัฒนธรรม แนวทางปฏิบัติ และเครื่องมือต่างๆ ที่ช่วยเพิ่มความสามารถขององค์กรในการส่งมอบแอปพลิเคชันและบริการอย่างรวดเร็ว โดยพัฒนาและปรับปรุงผลิตภัณฑ์ต่างๆ ให้เร็วกว่ากระบวนการการพัฒนาซอฟต์แวร์และการจัดการโครงสร้างพื้นฐานแบบดั้งเดิม ความรวดเร็วนี้ช่วยให้องค์กรสามารถให้บริการแก่ลูกค้าของตนได้ดีขึ้น และสามารถแข่งขันในตลาดได้อย่างมีประสิทธิภาพมากขึ้น

DevOps มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร

DevOps มีหลักการคือ การนำความคล่องตัวมาสู่งานการพัฒนาและการดำเนินงานระบบ **DevOps** อาจไม่สามารถทดแทน **Agile** ได้ แต่ **DevOps** จะสามารถนำมาปรับปรุงต่อยอดส่วนที่ยังขาดในเรื่องของการดำเนินงานได้ดี **DevOps** อาจไม่ใช่การแทนที่ **Agile** ได้ แต่ก็อาจใช้ร่วมเพื่อให้การปฏิบัติมีระบบดีขึ้น เมื่อเวลาผ่านไป **Agile** ก็ได้เพิ่มความท้าทาย และ **DevOps** ก็กลายเป็นแนวทางปฏิบัติที่เหมาะสมที่สุด

- กระบวนการหรือการปฏิบัติ ของ **DevOps** เกี่ยวข้องกับชุดของกระบวนการทางเทคนิคเช่น การพัฒนาอย่างต่อเนื่อง, การรวมอย่างต่อเนื่อง (CI), การทดสอบอย่างต่อเนื่อง (CI), การปรับใช้อย่างต่อเนื่อง (CD) และการตรวจสอบอย่างต่อเนื่อง
- พื้นที่โฟกัสที่สนใจให้ความสำคัญ จะมุ่งเน้นไปที่การรับประกันซอฟต์แวร์ที่มีคุณภาพในเวลาที่เหมาะสม การรับประกันคุณภาพจะทำการตรวจสอบซอฟต์แวร์อย่างต่อเนื่องหลังจากการปรับใช้
- มีรอบระยะเวลาทั้งในส่วนของการเผยแพร่และการพัฒนา จะมุ่งเน้นไปที่ระยะการปล่อยที่สั้นลง มุ่งให้มีการส่งมอบงานที่เร็วขึ้นแต่ติดตามผลการตอบรับอย่างต่อเนื่องทันที
- ผู้ให้ข้อเสนอแนะ จะมีการวัดผลจากทีมภายในพัฒนาภายใน โดยอาจใช้เครื่องมือการตรวจสอบมาช่วยอย่างต่อเนื่อง

CI คืออะไร

Continuous Integration (CI) คือ กระบวนการที่ใช้สำหรับการรวบรวมซอฟต์แวร์ที่มีการพัฒนาแยกส่วนกันอย่างอัตโนมัติ อาจจะโดยหนึ่งหรือหลายนักพัฒนาก็ตามที สุดท้ายแล้วซอฟต์แวร์ที่พัฒนาขึ้นเล็กๆ ที่พัฒนาขึ้นมาจะต้องนำมารวมกันเป็นชิ้นใหญ่หนึ่งชิ้น จะทำอะไรให้มั่นใจได้ว่า ไม่มีชิ้นส่วนใดที่จะส่งผลให้ชิ้นส่วนอื่นๆ พังเสียหาย เนื่องจากการพัฒนาโดยโปรแกรมเมอร์หลายคน

ซึ่งเป็นไปได้ว่าจะมี **bug** หลุดมาจากส่วนใดส่วนหนึ่ง แล้วเราจะป้องกันได้อย่างไรละ ดังนั้นจึงต้องมีการเขียน **script test** ที่คอยทดสอบความเข้ากันได้ของแต่ละชิ้นส่วนโดยอัตโนมัตินั่นเอง โดยการ **Testing** จะเริ่มตั้งแต่ **Unit Testing** ซึ่งสร้างจากทีมพัฒนา และเป็นส่วนจะใช้ตรวจสอบว่าสิ่งที่ทีมพัฒนายังทำงานถูกต้องและจะใช้เวลาช่วงสั้น ๆ เท่านั้น

โดยในโลกของการพัฒนานั้น มักใช้ **Build Server** มาช่วยเพื่อให้เป้าหมายที่ตั้งไว้สำเร็จ กล่าวคือ จะเริ่มทำการ **Integration** กันตั้งแต่เมื่อมีการเปลี่ยนแปลง **Source Code** ที่ **Repository** กลาง ระบบจะทำการตรวจสอบ **Code** หลังจากการเปลี่ยนแปลงว่าทำงานร่วมกันได้หรือไม่ตั้งแต่ **Compile, Testing**

CD คืออะไร

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

หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์จะต้องทำอย่างไรบ้าง ?

ขั้นตอนการพัฒนาระบบตามแนวทาง CI/CD

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

2. ทำการดึง source code ล่าสุดจาก Repository ของระบบ เพื่อตรวจสอบว่ามีการเปลี่ยนแปลงหรือไม่ถ้ามีการเปลี่ยนแปลงก็ให้ทำการรวมหรือ merge ที่เครื่องของ Developer ก่อน จากนั้นจึงทำการ build, test และ run อีกรอบเมื่อทุกอย่างผ่านทั้งหมด 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 ของเทสเคส ซึ่งเป็นชุดเทสเคสย่อยๆ 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 เกิดขึ้นจริงหรือไม่ถ้ามี bug ก็ให้ dev แก้ไข และ deploy มาใหม่ วง 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 ใหม่ๆ เพื่อ fixed bug จากที่ QA เจอ หรือที่พบเจอจากการ run

regression test แล้ว fail ก็จะเป็นการวน loop ตั้งแต่ต้นจนจบ จนกระทั่งทุกอย่างผ่านหมด Business พิจารณาว่าเอาขึ้น production ได้ เป็นการ confirm ว่าเราจะเอา code version สุดท้ายนี้ขึ้นไป production environment

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