

DevOps คือ [1]

DevOps เป็นแนวทางอีกแนวทางหนึ่งของระบบการพัฒนาซอฟต์แวร์ ที่ลดปัญหาการทำงานแบบเดิมที่ต้องเลือกระหว่างความเสถียรภาพกับความเร็วในการส่งมอบ ด้วยการค่อยๆส่งทีละส่วนทำให้ Ops ได้ทดสอบระบบได้บ่อยขึ้น ลูกค้าได้เห็นชิ้นงานได้ถี่ขึ้น Dev ก็สามารถรับ feedback ได้มากขึ้น ทำให้การพัฒนารวดเร็วแต่ก็มีความเสถียรภาพได้นั่นเองในกระบวนการผลิตซอฟต์แวร์จะแบ่งออกเป็น 2 ฝ่ายหลักคือ ฝ่าย Development กับฝ่าย Operations ซึ่งแต่ละฝ่ายมีหน้าที่คือ ฝ่าย Dev ต้องผลิตซอฟต์แวร์อย่างรวดเร็วส่งมอบให้ทันเวลา และฝ่าย Ops ต้องดูแลด้านความเสถียรของซอฟต์แวร์เพื่อมีความผิดพลาดน้อยที่สุด

ในกระบวนการการผลิตจะต้องเลือกเลือกระหว่าง [2]

1. การเลือกความเสถียรภาพ คือเมื่อทำการส่งมอบงาน งานที่ได้ออกมาจะมีจุดบกพร่องน้อยที่สุดและมีประสิทธิภาพและปลอดภัย ส่งมอบรอบเดียวจบ เมื่อเกิดปัญหาภายหลังตามมาจะได้ช่วยลดระยะเวลาในการแก้ไข แต่การเลือกแบบนี้มีข้อเสียคือใช้เวลานาน
2. การเลือกความเร็ว คือสามารถส่งมอบงานได้อย่างรวดเร็ว แต่ต้องแลกกับความเสถียรและประสิทธิภาพ โดยการส่งมอบงานจะมีโอกาสในการเกิดจุดผิดพลาดของงานที่มองไม่เห็น และอาจเป็นปัญหาใหญ่ตามมาทำให้เมื่อต้องการแก้ไขจะต้องใช้เวลานาน

DevOps เป็นแนวคิการทำงานที่เปลี่ยนรูปแบบระบบการทำงานของทีมโดยมีการเปลี่ยนแปลงที่สำคัญดังนี้ เปลี่ยนเป้าหมายการทำงานจากรับผิดชอบงานของตัวเองเป็นรับผิดชอบทั้งระบบ จากเดิมที่แต่ละฝ่ายมีเป้าหมายการทำงานไม่เหมือนกันเปลี่ยนให้กลายเป็นทีมเดียวกัน ทำงานร่วมกันมากขึ้น มีการรับผิดชอบร่วมกันมากขึ้น ทำให้ทั้ง 2 ฝ่ายสื่อสารกันเข้าใจปัญหาของแต่ละฝ่าย ทุกฝ่ายเข้าใจภาพรวมของงานมากขึ้นทำให้ปัญหา และหน้าที่การแก้ปัญหาไม่ตกไปอยู่ฝ่ายใดฝ่ายหนึ่งเพียงฝ่ายเดียว เช่นฝ่าย Dev เข้าใจว่าสิ่งที่ผู้ใช้ต้องการจริงๆคืออะไร ไม่ใช่ทำตามเพียงคำสั่งที่ได้รับเข้ามา ฝ่าย Ops เข้าใจการทำงานของโปรแกรมมากขึ้นสามารถรู้ได้ทันทีว่าปัญหาของผู้ใช้เกิดขึ้นที่ตรงไหน

เปลี่ยนจากการส่งมอบแบบรอบเดียวจบต่อ 1 งาน เป็นส่งเป็นรอบย่อย ๆ ใน 1 งาน จากเดิมที่ทั้ง 2 ทีมทำหน้าที่กันคนละส่วน งาน 1 งาน Dev ทำงานส่วนของตัวเองเสร็จจึงส่งมอบให้ Ops ไปจัดการติดตั้งแล้วทดสอบต่อ ทำให้กระบวนการติดตั้งและทดสอบหลักๆสามารถเกิดขึ้นระหว่างการพัฒนา 1 ครั้งเท่านั้น ตอนนี้อย่างน้อยต้องเปลี่ยนใหม่เป็น งาน 1 งานย่อยเป็นหลายส่วน เมื่อทำเสร็จส่วนย่อย ๆ 1 ส่วน ก็นำไปติดตั้งและทดสอบทันที ทำให้ได้ทดสอบหลายครั้ง ทดลองตั้งค่าหลายครั้ง และได้รับ feedback ทันทีในแต่ละส่วนๆ ทำให้ตรงกับความต้องการจริงๆ และไม่เป็นปัญหาที่มีจำนวนมากและกลายเป็นปัญหาที่ใหญ่ตามมา

นอกจากระบบการทำงานดีแล้วเครื่องมือก็เป็นสิ่งสำคัญ

จากเดิมที่แต่ละฝ่ายทำงานแยกกัน ตอนนี้หันมาทำงานร่วมกันแล้ว แต่ก็ยังมีปัญหาที่ไม่สามารถแก้ได้ด้วยระบบการทำงานแต่สามารถใช้เครื่องมือมาช่วยให้การทำงานดีขึ้นได้ เช่น ปัญหาที่ว่าทดสอบระหว่างพัฒนามันรัน/ติดตั้งได้แต่ตอนที่ให้คนอื่นทดสอบทำไมมันรัน/ติดตั้งไม่ได้ ปัญหานี้เกิดจากสภาพแวดล้อมของการทำงาน โปรแกรมของแต่ละเครื่องนั้นอาจไม่เหมือนกัน ทำให้โปรแกรมทำงานต่างกัน เราสามารถใช้เครื่องมืออย่าง container ที่จำลองสภาพแวดล้อมที่เหมาะสมให้กับโปรแกรมมาช่วยตรงจุดนี้ สภาพแวดล้อมในระหว่างที่โปรแกรมทำงานของผู้พัฒนากับผู้ทดสอบหรือผู้ใช้เหมือนกันแน่นอน สามารถไปรันที่ไหนก็ได้ส่งมอบได้อย่างไม่มีปัญหา

CI / CD คือ [3]

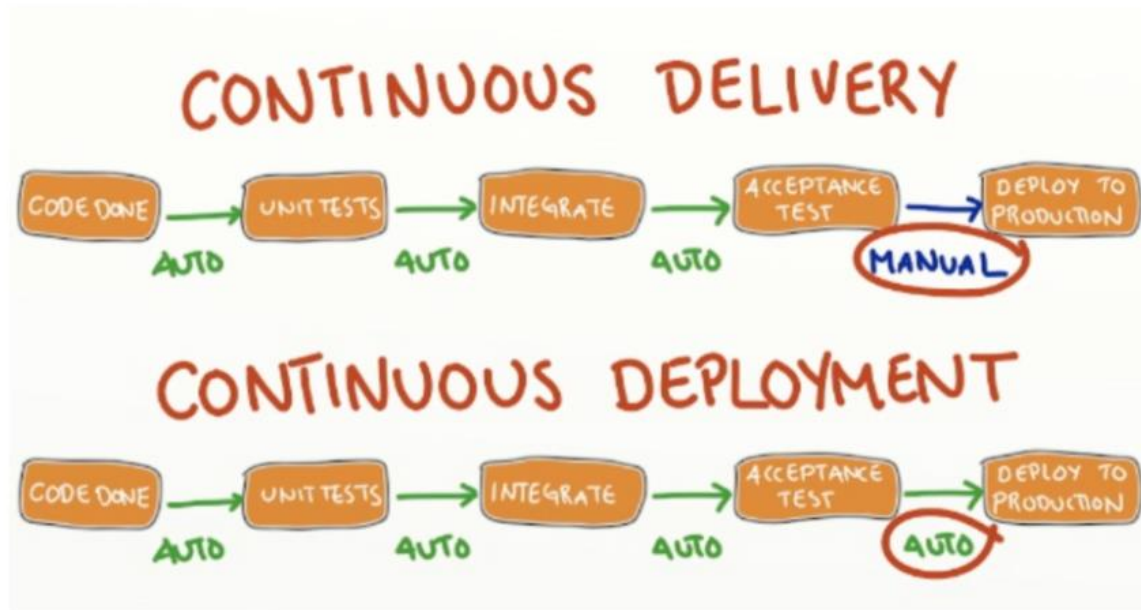
Continuous Integration (CI) คือ กระบวนการใช้สำหรับการรวบรวมซอฟต์แวร์ที่มีการพัฒนาแยกส่วนกัน อย่างอัตโนมัติ อาจจะโดยหนึ่งหรือหลายนักพัฒนาก็ตามที สุดท้ายแล้วซอฟต์แวร์ที่พัฒนาขึ้นเล็กๆ ที่พัฒนาขึ้นมาจะต้องนำมารวมกันเป็นชิ้นใหญ่หนึ่งชิ้น จะทำอะไรให้มั่นใจได้ว่า ไม่มีชิ้นส่วนใดที่จะส่งผลให้ชิ้นส่วนอื่นๆ พังเสียหาย เนื่องจากการพัฒนาโดยโปรแกรมเมอร์หลายคน ซึ่งเป็นไปได้ว่าอาจจะมี bug หลุดมาจากส่วนใดส่วนหนึ่งดังนั้นจึงต้องมีการเขียน script test ที่คอยทดสอบความเข้ากันได้ของแต่ละชิ้นส่วนโดยอัตโนมัตินั่นเอง โดยการ Testing จะเริ่มตั้งแต่ Unit Testing ซึ่งสร้างจากทีมพัฒนา และเป็นส่วนจะใช้ตรวจสอบว่าสิ่งที่ทีมพัฒนายังทำงานถูกต้องและจะใช้เวลาช่วงสั้น ๆ เท่านั้น โดยในการพัฒนานั้น มักใช้ Build Server มาช่วยเพื่อให้เป้าหมายที่ตั้งไว้สำเร็จ กล่าวคือ จะเริ่มทำการ Integration กันตั้งแต่เมื่อมีการเปลี่ยนแปลง Source Code ที่ Repository กลาง ระบบจะทำการตรวจสอบ Code หลังจากการเปลี่ยนแปลงว่าทำงานร่วมกันได้หรือไม่ตั้งแต่ Compile, Testing

CD จะมี 2 ตัวคือ

Continuous Deployment คือ ในทุกๆ ขั้นตอนจนถึงการ deployment ขึ้น production จะทำแบบอัตโนมัติทั้งหมด

Continuous Delivery คือ การงานต่างๆ ใน deployment pipeline นั้น จะเริ่มต้นทำงานตั้งแต่การ compile, build ไปจนถึงขั้นตอนการทดสอบต่างๆ เช่น Acceptance test เป็นแบบอัตโนมัติทั้งหมด ส่วนในขั้นตอนการ

deployment ขึ้น production นั้น จะต้องได้รับการอนุมัติหรือการตัดสินใจกันก่อนจากทาง Business ซึ่งเป็นการทำงานแบบ manual



(ที่มา : https://miro.medium.com/max/3000/1*4jFpFuKaHVRZ2xUrTiisCA.png)

ซึ่งส่วนของการทำชุดทดสอบระบบนั้น QA จะทำการ เลือกชุด test case ขึ้นมา สำหรับแต่ละ component หรือ service ย่อย ๆ การเพิ่มเติมคือเขียน script automated testing ในที่นี้ใช้ Git ผ่าน Source Tree ในการจัดการ source code automated test case และ ใช้ Jenkins ในการ config ให้ Trigger สั่งให้ Run ชุดทดสอบ automated testing ในแต่ละ environment

Automated testing ถูกนำไปใช้ใน Process ของ CI/CD ทุกการเปลี่ยนแปลงหนึ่งนั้นคือการทดสอบนั่นเอง แน่นอนว่า ต้องเป็นการทดสอบแบบอัตโนมัติอย่างแน่นอน (Automated testing) ไม่ว่าจะเป็น Unit testing, Integration testing และ Functional testing

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 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 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 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 ผ่านหมด Business ฟันธง มาว่าเอาขึ้น production ได้ เป็นการ confirm ว่าเราจะเอา code version สุดท้ายนี้ ขึ้นไปที่ production environment

ข้อดีของการพัฒนาระบบตามแนวทาง CI/CD

ลดระยะเวลาในการพัฒนา ลดต้นทุนในการพัฒนา รองรับการเปลี่ยนแปลงแก้ไขได้ง่าย ดูแลรักษาได้ง่าย เนื่องจากการดำเนินไปอย่างต่อเนื่อง เป็นไปอย่างอัตโนมัติและทราบผลลัพธ์ได้ทันที ทำให้เราสามารถส่งมอบ แอปฯ รุ่นใหม่ให้กับลูกค้าได้ไวขึ้น

โดยทั้งสองมีสัมพันธ์กันคือเป็นการพัฒนาซอฟต์แวร์ให้มีประสิทธิภาพเพิ่มมากขึ้น การทำงานมีแบบแผน และสามารถส่งมอบได้โดย

สามารถนำแนวทางการปฏิบัติงานของ DevOps ซึ่งเป็นการทำงานที่ร่วมกัน มีการวางแผนอย่างหลักการ และการสื่อสารมีการใช้เครื่องมือที่เหมาะสม มาผสมกับ CI / CD ซึ่งเป็นเครื่องมือที่ช่วยในการทำการทดสอบแบบอัตโนมัติที่เหมาะสม ซึ่งสามารถทำให้ ซอฟต์แวร์มีประสิทธิภาพและสามารถสร้างซอฟต์แวร์ให้ดีขึ้นโดยใช้เวลาน้อยลง

ยกตัวอย่างเช่นมีบริษัทหนึ่งต้องการสร้างเว็บสำหรับการซื้อขายสินค้า ซึ่งในการซื้อขายสินค้านั้นมีรายละเอียดที่ซับซ้อนมาก ทำให้ต้องมีการวางแผนออกมาอย่างเป็นระบบระเบียบและในทุกฟีเจอร์การทำงานจะต้องมีการทดสอบซึ่งการทำ CI / CD จะเข้ามาช่วยในด้านนี้โดยจะเป็นการทดสอบระบบแบบอัตโนมัติเพื่อช่วยประหยัดเวลาในทดสอบระบบและสำหรับ DevOps จะเข้ามาช่วยในการวางแผนงานเพื่อให้นักพัฒนาโปรแกรมเมอร์เกิดประสิทธิภาพมากที่สุด เมื่อส่งมอบแล้วเวลาเกิดปัญหาจะได้ไม่เป็นปัญหาที่ใหญ่ ซึ่งเมื่อกระทำการตามนี้แล้วจะช่วยให้งานนั้นมีประสิทธิภาพเพิ่มมากขึ้นและลดระยะเวลาการทำงานได้มากแน่นอน

อ้างอิง

- [1] Developer. DevOps คืออียังวะ ? | BorntoDev Digital Academy [Internet]. BorntoDev เริ่มต้นเรียนเขียนโปรแกรม ขั้นเทพ2020 [cited 2021 Feb 25];Available from: <https://www.borntodev.com/2020/05/15/devops-คืออียังวะ/>
- [2] DevOps คืออะไร - Amazon Web Services (AWS) [Internet]. Amaz. Web Serv. Inc [cited 2021 Feb 25];Available from: <https://aws.amazon.com/th/devops/what-is-devops/>
- [3] narissara. การนำ Automated testing มารวมเข้ากับ กระบวนการ CI/CD [Internet]. Medium2017 [cited 2021 Feb 25];Available from: <https://medium.com/@narissara.tan/การนำ-automated-testing-มารวมเข้ากับ-กระบวนการ-ci-cd-58c97e31e2300>