

### 3. ศึกษาเกี่ยวกับ DevOps และ CI/CD ให้เข้าใจ

#### 4. เขียนสรุปเกี่ยวกับ DevOps และ CI/CD ลงในไฟล์ชนิด .doc, docx, หรือ .pdf ไม่เกิน 6 หน้ากระดาษ ให้ครอบคลุมประเด็นดังต่อไปนี้

- (1) DevOps คืออะไร มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร
- (2) CI/CD คืออะไร มีความสัมพันธ์กับ DevOps อย่างไร
- (3) หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมใน

กระบวนการพัฒนาซอฟต์แวร์จะต้องทำอย่างไรบ้าง

ใส่แหล่งอ้างอิงให้ถูกต้องตามหลักวิชาการ เก็บไฟล์ดังกล่าวไว้ใน Folder ชื่อ DevOps ที่สร้างในข้อที่ 2

DevOps เป็นคำใหม่ที่เอาคำว่า Developer กับ Operator มารวมกัน (1) โดยสมัยก่อน Developer เป็นคนพัฒนา code แต่ไม่มีสิทธิ deploy code ทำให้การทำงานยุ่งยาก เกิดความผิดพลาด และใช้เวลานานในการ deploy แต่ละครั้ง เพื่อแก้ปัญหาในการ deploy code จึงนำเอาทั้งสองทีมมาทำงานร่วมกันเกิดเป็นแนวทาง CI/CD โดยทำงานตั้งแต่การ Plan, Code, Build, Test, Release, Deploy, Operate, Monitor หรือบางที่เรียกสั้นๆ ว่า Pipeline

#### ประโยชน์ของ DevOps (2)

ความเร็ว ดำเนินการอย่างรวดเร็วเพื่อให้สามารถสร้างสรรค์นวัตกรรมให้แก่ลูกค้าได้รวดเร็วยิ่งขึ้น ปรับตัวต่อการเปลี่ยนแปลงของตลาดได้ดียิ่งขึ้น และมีประสิทธิภาพยิ่งขึ้นในการกระตุ้นให้เกิดผลลัพธ์ทางธุรกิจ

การส่งมอบอย่างรวดเร็ว ออกรุ่นใหม่ ๆ ที่ขึ้นและรวดเร็วยิ่งขึ้น เพื่อให้สามารถสร้างสรรค์นวัตกรรมและปรับปรุงผลิตภัณฑ์ได้รวดเร็วยิ่งขึ้น ยังสามารถออกคุณสมบัติใหม่ ๆ และแก้ไขจุดบกพร่องได้เร็วเท่าไร ก็ยิ่งตอบสนองความต้องการของลูกค้าและสร้างความได้เปรียบในการแข่งขันได้เร็วเท่านั้น

ความเชื่อถือได้ ทำให้แน่ใจถึงคุณภาพของการอัปเดตแอปพลิเคชันและการเปลี่ยนแปลงโครงสร้างพื้นฐาน เพื่อให้สามารถส่งมอบได้อย่างน่าเชื่อถือในความเร็วที่มากขึ้น ในขณะที่ยังคงรักษาประสิทธิภาพใช้งานเชิงบวกของผู้ใช้ปลายทาง ใช้ข้อปฏิบัติอย่างเช่น การบูรณาการอย่างต่อเนื่อง และการส่งมอบอย่างต่อเนื่อง เพื่อทดสอบว่าการเปลี่ยนแปลงนั้นทำงานได้และปลอดภัยหรือไม่

ขนาด ดำเนินการและจัดการโครงสร้างพื้นฐานและกระบวนการพัฒนาในขนาดต่างๆ การทำงานอัตโนมัติและความสอดคล้องจะช่วยให้จัดการระบบที่ซับซ้อนหรือมีการเปลี่ยนแปลงได้อย่างมีประสิทธิภาพ โดยมีความเสี่ยงที่ลดลง

การทำงานร่วมกันที่ปรับปรุงดีขึ้น สร้างทีมงานที่มีประสิทธิผลมากขึ้นภายใต้โมเดลเชิงวัฒนธรรมของ

DevOps เช่น นักพัฒนาและทีมปฏิบัติการจะทำงานร่วมกันอย่างใกล้ชิด รับผิดชอบร่วมกันในหลายกรณี และนำลำดับการทำงานมารวมเข้าด้วยกัน ซึ่งจะช่วยลดความไร้ประสิทธิภาพและประหยัดเวลา การรักษา

ความปลอดภัย ดำเนินการอย่างรวดเร็วในขณะที่ยังรักษาการควบคุมและการปฏิบัติสอดคล้องกับกฎระเบียบ สามารถรับเอาโมเดล DevOps มาใช้โดยไม่ต้องสูญเสียความปลอดภัยโดยใช้นโยบายการปฏิบัติตามกฎระเบียบที่ทำงานอัตโนมัติ การควบคุมโดยละเอียด และเทคนิคการจัดการการกำหนดค่า

CI/CD คือ กระบวนการหนึ่งที่จะมาช่วยในพัฒนา Software ให้มีประสิทธิภาพมากขึ้น ทั้งในแง่ของระยะเวลาการพัฒนาและคุณภาพของ Software (3) มีความสัมพันธ์กับ DevOps โดยแก้ปัญหาการ deploy code โดยนำ CI/CD มาปรับใช้ (4) Continuous Integration (CI) คือ กระบวนการที่ใช้สำหรับการรวบรวมซอฟต์แวร์ที่มีการพัฒนาแยกส่วนกันอย่างอัตโนมัติ (3) ซอฟต์แวร์ชิ้นเล็กๆ ที่พัฒนาขึ้นมาจะต้องนำมารวมกันเป็นชิ้นเดียว แล้วเราจะสามารถมั่นใจได้อย่างไรว่า ชิ้นส่วนชิ้นหนึ่งไม่มีผลกระทบทำให้ชิ้นส่วนชิ้นอื่นๆ เสียหาย แล้วเราจะสามารถป้องกันได้อย่างไร ดังนั้นจึงต้องมีการเขียน script test ที่คอยทดสอบความเข้ากันได้ของแต่ละชิ้นส่วนโดยอัตโนมัตินั่นเอง โดยการ Testing จะเริ่มตั้งแต่ Unit Testing ซึ่งสร้างจากทีมพัฒนา และเป็นส่วนจะใช้ตรวจสอบว่าสิ่งที่ทีมพัฒนายังทำงานถูกต้องและจะใช้เวลาช่วงสั้น ๆ เท่านั้น มักใช้ Build Server มาช่วยเพื่อให้เป้าหมายที่ตั้งไว้สำเร็จ กล่าวคือ จะเริ่มทำการ Integration กันตั้งแต่เมื่อมีการเปลี่ยนแปลง Source Code ที่ Repository กลาง ระบบจะทำการตรวจสอบ Code หลังจากการเปลี่ยนแปลงว่าทำงานร่วมกันได้หรือไม่ตั้งแต่ Compile, Testing ส่วน Continuous Delivery หรือ Continuous Deployment (CD) คือ กระบวนการที่ช่วยให้เราสามารถ Deploy Software ของเราได้อย่างมีประสิทธิภาพ (3) โดยการนำ Source Code ที่ผ่านการ Build และ Testing มาแล้ว ซึ่งอาจอยู่ในรูปแบบที่แตกต่างกัน เช่น JAR file, Static file หรือแม้กระทั่ง Container Image ให้จัดการ Deploy ขึ้นไปอยู่บน Server ตามที่เราต้องการและสามารถใช้งานได้อย่างถูกต้อง

กระบวนการ CD มี 2 ประเภท

Continuous Delivery คือ การส่งมอบ Software ที่ใช้วิธีการ manual ในการ Deploy หลังจากผ่านขั้นตอน CI มาเรียบร้อยแล้ว โดยจะต้องมีการอนุมัติจากผู้รับรอง ซึ่งในที่นี้อาจหมายถึง Manager เพื่อทำการตรวจสอบก่อน Deploy ขึ้นสู่ Server การทำงานของ CD ตามปกติคือ (5) เมื่อเรา push งานขึ้นมาที่เซิร์ฟเวอร์แล้ว ระบบก็จะนำมา build test ต่างๆตามสิ่งที่เรากำหนดไว้ใน Continuous Integration (CI) ซึ่งถ้ามัน build ผ่านหมดแล้ว เราก็จะให้มันทำการ deploy ลงไปที่เซิร์ฟเวอร์แบบอัตโนมัตินั่นเอง แต่โดยปกติเราจะ deploy ไปที่ตัว Test Environment หรือ Staging Environment เพื่อตรวจสอบเช็คความถูกต้องต่างๆ ก่อน เช่นดูว่า features ใหม่ถูกต้องไหม UX ดีพอหรือยัง ฯ ซึ่งเมื่อเราทดสอบทุกอย่างแล้ว เราก็จะทำการส่ง CD ไปยัง Production Environment ในขั้นตอนสุดท้าย

Continuous Deployment คือ การส่งมอบ Software ในรูปแบบ Automate หลังจากผ่านขั้นตอน CI มาเรียบร้อยแล้ว จะทำการ Deploy ขึ้นสู่ Server ทันที ไม่ต้องรอการอนุมัติใด ๆ

ประโยชน์ของการพัฒนาระบบตามแนวทาง CI/CD (4)

ลดระยะเวลาในการพัฒนา ลดต้นทุนในการพัฒนา รองรับการเปลี่ยนแปลงแก้ไขได้ง่าย ดูแลรักษาได้ง่าย  
เนื่องจากการดำเนินการไปอย่างต่อเนื่อง เป็นไปอย่างอัตโนมัติและทราบผลลัพธ์ได้ทันที ทำให้เราสามารถส่งมอบแอปฯ รุ่นใหม่ให้กับลูกค้าได้ไวขึ้น

หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์จะต้องทำอย่างไรบ้าง

DevOps นั้นมีรูปแบบการทำงานค่อนข้างจะตายตัว ซึ่งจะพยายามทำให้ Process ต่างๆ ทำงานไปได้โดยอัตโนมัติ ไม่ต้องมีคนคอยเข้าไปกด Deploy เองอีกต่อไป ยกเว้นในส่วนของการวางแผน การพัฒนา และดูแลผลลัพธ์ (6)

Plan การจะพัฒนาโปรแกรมขึ้นมา ต้องมีการวางแผนร่วมกันจากทุกฝ่ายที่เกี่ยวข้องอย่างละเอียดก่อน โปรแกรมเราจะทำอะไรบ้าง จะพัฒนาด้วยภาษาอะไร Framework อะไร จะรันบน Platform อะไร จะไปลงไว้ที่ไหน จะเก็บข้อมูลอย่างไร จะแก้ปัญหาที่เกิดขึ้น และจะพัฒนา Process การทำงานอย่างไร

Create การสร้างหรือการพัฒนา นี่ก็คือขั้นตอนของการเขียนโปรแกรมขึ้นมา DevOps เข้ามาช่วยให้ Developer ทุกคนทำงานบน Environment แบบเดียวกัน ไม่ว่าเขาจะพัฒนาบน OS ใดก็ตาม เวลา Developer ทำงานบนคอมพิวเตอร์คนละ OS กัน มักจะมีปัญหา “ทำไมโปรแกรมรันบนเครื่องเธอไม่ได้ละ บนเครื่องฉันก็รันได้อยู่นะ!” ซึ่งมันไม่ควรจะเกิดขึ้นแล้วในยุคนี้ที่เรามี Docker ที่ช่วยให้ทุกคนพัฒนาโปรแกรมอยู่บนพื้นฐาน Environment เดียวกันเป๊ะ ตั้งแต่ขั้นตอนการพัฒนาไปจนถึงการส่งโปรแกรมขึ้น Production

Verify การตรวจสอบอยู่อย่างสม่ำเสมอว่าโปรแกรมที่พัฒนาขึ้นมา ไม่ว่าจะเพิ่มฟีเจอร์อะไรเข้ามา จะไม่ไปกระทบหรือไปก่อบั๊กทำให้การใช้งานเดิมพังเสียหายนั้นเป็นเรื่องสำคัญมาก โดยเฉพาะโปรแกรมขนาดใหญ่ที่ร่วมกันทำกับทีมงานจำนวนมาก เราจะมั่นใจได้อย่างไรว่า Code ที่ Developer X ส่งขึ้นมาจะเข้ากันได้ดีกับการทำงานเดิม

วิธีการที่จะรู้ได้เร็วที่สุดคือ เราจะต้องมีการทำ Testing อย่างครอบคลุมในทุกๆ Components ของโปรแกรม แล้วทำให้มันรันโดยอัตโนมัติทุกครั้งก่อนจะปล่อยโปรแกรมออกไป เมื่อใดที่ Test Fail ก็จะได้รู้ได้ทันทีว่ามีปัญหาเกิดขึ้นแล้ว ให้หยุดการส่งงานที่มีปัญหาไปก่อน สามารถเข้าไปหาสาเหตุและแก้ไขปัญหาก็ได้อย่างทันท่วงที

Package การจะส่งโปรแกรมขึ้นไปรันบน Server เพื่อความสะดวกรวดเร็ว จะต้องมีการเอาโปรแกรมไปใส่ใน Technology ที่ออกแบบมาเพื่อนำโปรแกรมไปรันได้อย่างราบรื่น ซึ่งในปัจจุบันนี้เราใช้

Docker กัน เวลาจะใช้ เราจะสร้าง Docker Image ที่เอาตัวโปรแกรมพร้อมสำหรับการรันใส่อยู่ในนั้น แล้วเอา Docker Image นี้ไป Deploy ที่ไหนก็ได้สามารถรัน Docker ได้ โปรแกรมก็จะรันขึ้นมาพร้อมใช้งาน

Release เมื่อมี Docker Image พร้อมแล้ว ก็พร้อมสำหรับการนำโปรแกรมไปรันบน Deployment Platform ที่ต้องการ ตอนที่กำลังทำของ Development หรือทำหรือ Production อยู่ จะต้องเลือกไป Deploy ให้ถูกที่

Configure โปรแกรมที่พัฒนานั้น จำเป็นอย่างมาก ที่จะต้องตั้งค่าได้ ตัวอย่างการตั้งค่าที่ควรมีเลยก็เช่น การเลือก Database ตั้งค่า Email เป็นต้น ยิ่งถ้ามี Service แยกอื่นๆ ที่ต้องใช้แล้ว ยิ่งสำคัญ เพราะ Development Server และ Production Server ควรอยู่แยกกันอย่างเด็ดขาด เราจะต้องหาทางสร้าง Docker Image มา 1 อัน แล้วเราจะใช้มัน Deploy ลงบน Environments ที่ต่างกันให้ได้ วิธีการที่จะทำได้ก็คือการทำให้โปรแกรมของเราตั้งค่าได้เนี่ยแหละ คือโปรแกรมที่ไปรันเป็นตัวเดียวกัน แต่เปิดด้วยการตั้งค่าที่แตกต่างกัน ทำให้ได้การทำงานที่แตกต่างกัน โดยปกติจะใช้ไฟล์ .env และ Environment Variable ในการตั้งค่ากันเป็นมาตรฐานที่ใช้กันบนทุก Platform

Monitor สุดท้ายที่ขาดไม่ได้เลยคือการตรวจสอบว่าโปรแกรมมันขึ้นไปแล้วทำงานได้ปกติ มีสุขภาพแข็งแรงดี การดู Log ว่าโปรแกรมเราทำงานมีปัญหาตรงไหน มีความสำคัญอย่างมากในการแก้ไข บั๊ก ยิ่งในตอนนี้ที่มักจะออกแบบให้โปรแกรมมันสามารถทำให้ Horizontal Scale ได้ (การสร้างเครื่องขึ้นมาหลายๆ เครื่องให้มาแบ่ง Load ทำงานแยกกันได้) จำเป็นอย่างยิ่งที่จะต้อง มี Centralized Logging System รวบรวม Log เอามาไว้ในที่เดียว เพราะการ Remote เข้าไปอ่าน Log ในแต่ละเครื่องเป็นไปไม่ได้ อีกแล้ว

อีกส่วนหนึ่งที่สำคัญไม่แพ้กันคือการทำ Monitoring ดูโหลดต่างๆ ของระบบ ตอนนี้มีก็เครื่องเปิดขึ้นมาใช้งาน เครื่องล่มหรือเปล่า ให้บริการได้ทันทีลูกค้าเข้ามาใช้หรือไม่ เรื่องเหล่านี้ล้วนมีเครื่องมือออกมาแก้ไขอยู่แล้วทั้งสิ้น เพียงแต่ต้อง Implement โปรแกรมให้ไปรองรับการใช้งานกับเครื่องมือเหล่านั้น

ตัวอย่างขั้นตอนการพัฒนาระบบตามแนวทาง CI/CD (4)

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

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 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 เกิดขึ้นจริงหรือไม่ ถ้ามี 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 ซ้ำๆ เพื่อ fixed bug จากที่ QA เจอ หรือที่พบเจอจากการ run regression test แล้ว fail ก็จะเป็นการวน loop ตั้งแต่ต้นจนจบ จนกระทั่ง ทุกอย่างผ่านหมด Business พิจารณาว่าเอาขึ้น production ได้ เป็นการ confirm ว่าเราจะเอา code version สุดท้ายนี้ขึ้นไปที่ production environment

## อ้างอิง

1. DevOps CI/CD คืออะไร? – Ton's Blog [อินเทอร์เน็ต]. [อ้างถึง 26 กุมภาพันธ์ 2021]. Available at: <https://ton.packetlove.com/blog/life-style/devops-ci-cd.html>
2. DevOps คืออะไร - Amazon Web Services (AWS) [อินเทอร์เน็ต]. [อ้างถึง 26 กุมภาพันธ์ 2021]. Available at: <https://aws.amazon.com/th/devops/what-is-devops/>
3. ทำความรู้จักกับ CI/CD services บน Google Cloud [อินเทอร์เน็ต]. [อ้างถึง 26 กุมภาพันธ์ 2021]. Available at: <https://cloud-ace.co.th/blog/w0w7e1-ci-cd-services-google-cloud>
4. การพัฒนาระบบตามแนวทาง CI/CD และ DevOps คืออะไร? [อินเทอร์เน็ต]. [อ้างถึง 26 กุมภาพันธ์ 2021]. Available at: <https://www.softmelt.com/article.php?id=664>
5. ลองทำ Continuous Delivery (CD) - Saladpuk.com [อินเทอร์เน็ต]. [อ้างถึง 26 กุมภาพันธ์ 2021]. Available at: <https://www.saladpuk.com/cloud/azure-devops/cd>

6. DevOps คืออะไร นำมาประโยชน์ได้ยังไง และตัวอย่างการทำ DevOps ที่ Credit OK | SPICYDOG's Blog [อินเทอร์เน็ต]. [อ้างถึง 26 กุมภาพันธ์ 2021]. Available at:  
<https://www.spicydog.org/blog/introduction-to-devops-and-the-practical-use-cases-at-credit-ok/>