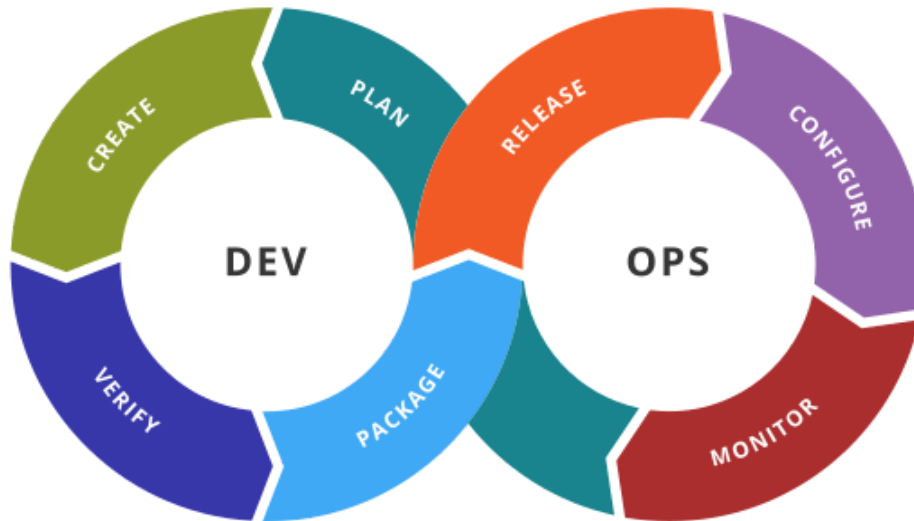


Assignment#1 วิชา SC313004 Software Engineering - DevOps and CI/CD

1. DevOps คืออะไร มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร

Ans.



source : <https://www.redmineup.com/pages/blog/devops-in-redmine>

ความหมายของ DevOps

ในกระบวนการผลิตซอฟต์แวร์ DevOps มาจากคำว่า Development กับ Operations เมื่อก่อน 2 อย่างนี้เป็นหน้าที่หลักในการผลิตซอฟต์แวร์ ฝ่าย Dev จะเข้าใจสิ่งที่ผู้ใช้ต้องการอะไรจริงๆ และผลิตซอฟต์แวร์ให้ทันเวลาและรวดเร็ว ส่วนฝ่าย Ops จะเข้าใจการทำงานของโปรแกรมที่สามารถรู้ได้ว่าปัญหาของผู้ใช้เกิดขึ้นในส่วนไหนและดูแลความเสถียรของซอฟต์แวร์ให้มีความผิดพลาดน้อยที่สุด ซึ่งทั้งสองฝ่ายนี้มีหน้าที่เป้าหมายที่เหมือนกันคือใช้เครื่องมือที่แตกต่างกัน จึงทำให้เกิดข้อขัดแย้งต่างๆ ซึ่ง DevOps นี้เป็นแนวคิดที่จะมาช่วยลดข้อขัดแย้งที่เกิดขึ้น

สรุปแล้ว **DevOps** หมายถึง แนวทางของระบบการพัฒนาซอฟต์แวร์ที่ช่วยแก้ไขปัญหาแบบเดิมๆ ช่วยเชื่อมให้การพัฒนาโปรแกรม เครื่องมือ การทำงานต่างๆ ให้มีประสิทธิภาพในการทำงานให้สามารถผลิตซอฟต์แวร์ให้กับผู้ใช้ได้อย่างรวดเร็ว มีคุณภาพและเสถียรภาพมากขึ้น รวมทั้งการติดตามสถานะการทำงานของโปรแกรมเพื่อนำกลับมาพัฒนาปรับปรุงกระบวนการพัฒนาโปรแกรมให้ดีขึ้น และเพื่อทำ Automation ด้วย

DevOps Process

1. **Plan** การวางแผนร่วมกันอย่างละเอียดก่อนว่าจะทำอะไรบ้าง ด้วยภาษาอะไร ที่ไหน อย่างไร
2. **Create** สร้างหรือพัฒนา เขียนโปรแกรมขึ้นมา โดยใช้เครื่องมืออย่าง Docker เพื่อให้ Environment เหมือนกันและทำงานร่วมกันได้
3. **Verify** ตรวจสอบอย่างสม่ำเสมอ ว่ามีอะไรเพิ่มเข้ามาที่จะทำให้เกิด bug หรือไม่เพื่อไม่ให้เกิดความเสียหาย หรือ Test บ่อยๆ ทำให้มั่นใจระบบอัตโนมัติ ถ้ามีปัญหาจะได้แก้ทัน
4. **Package** ส่งโปรแกรมขึ้นไปรันบน server เพื่อสะดวกและรวดเร็ว โดยเอาโปรแกรมไปใส่ Docker แล้ว Deploy ที่เซิร์ฟเวอร์ที่ไหนก็ได้
5. **Release** เมื่องานพร้อมแล้วก็นำไป Deployment Platform ที่ต้องการ

6. **Configure** โปรแกรมที่พัฒนาต้องตั้งค่าได้ โปรแกรมที่รันเป็นตัวเดียวกันแต่ตั้งค่าแตกต่างกันแต่ต้องเป็นมาตรฐานทุก platform
7. **Monitor** ตรวจสอบว่าโปรแกรมทำงานได้ปกติ ดูว่ามีปัญหาตรงไหน ดูโหลดต่างๆของระบบ มีกี่เครื่องที่เปิดใช้งาน เครื่องล่มหรือไม่ ให้บริการทันหรือไม่ implement โปรแกรมให้รองรับกับเครื่องมือต่างๆ

ประโยชน์ของ DevOps

1. **ความเร็ว** ดำเนินการอย่างรวดเร็วเพื่อให้สร้างสรรค์งานใหม่ๆให้แกลูกค้าได้เร็วขึ้น ปรับตัวต่อการเปลี่ยนแปลงได้ดีและมีประสิทธิภาพที่ช่วยกระตุ้นผลลัพธ์ทางธุรกิจ
2. **การส่งมอบอย่างรวดเร็ว** มีรุ่นใหม่ออกมาเรื่อยๆ เพื่อให้สร้างสรรค์งานใหม่ๆและปรับปรุงผลิตภัณฑ์ได้รวดเร็ว ทำให้สามารถแก้ไขจุดบกพร่องได้ทันและตอบสนองตามผู้ใช้ได้
3. **ความเชื่อถือได้** ส่งมอบได้เร็วและน่าเชื่อถือ ทำให้แน่ใจถึงคุณภาพของการอัปเดตแอปพลิเคชันและการเปลี่ยนแปลงโครงสร้างพื้นฐาน ที่ยังคงรักษาการใช้งานของผู้ใช้ปลายทางได้
4. **ขนาด** ดำเนินการและจัดการโครงสร้างพื้นฐาน กระบวนการพัฒนาในขนาดต่างๆ การทำงานอัตโนมัติและความสอดคล้องจะช่วยให้จัดการระบบที่ซับซ้อนได้อย่างมีประสิทธิภาพ
5. **การทำงานร่วมกันที่ปรับปรุงดีขึ้น** สร้างทีมที่มีประสิทธิผลมากขึ้น เช่น นักพัฒนาและทีมปฏิบัติการร่วมกัน รับผิดชอบงานร่วมกัน และนำลำดับการทำงานมารวมกัน ช่วยลดความไร้ประสิทธิภาพและประหยัดเวลา
6. **การรักษาความปลอดภัย** ดำเนินการอย่างรวดเร็วและยังคงรักษาการควบคุมและการทำงานให้สอดคล้องกับกฎระเบียบ สามารถเอาโมเดลมาใช้โดยไม่ต้องสูญเสียความปลอดภัยโดยใช้ตามกฎหมายที่ทำงานอัตโนมัติ

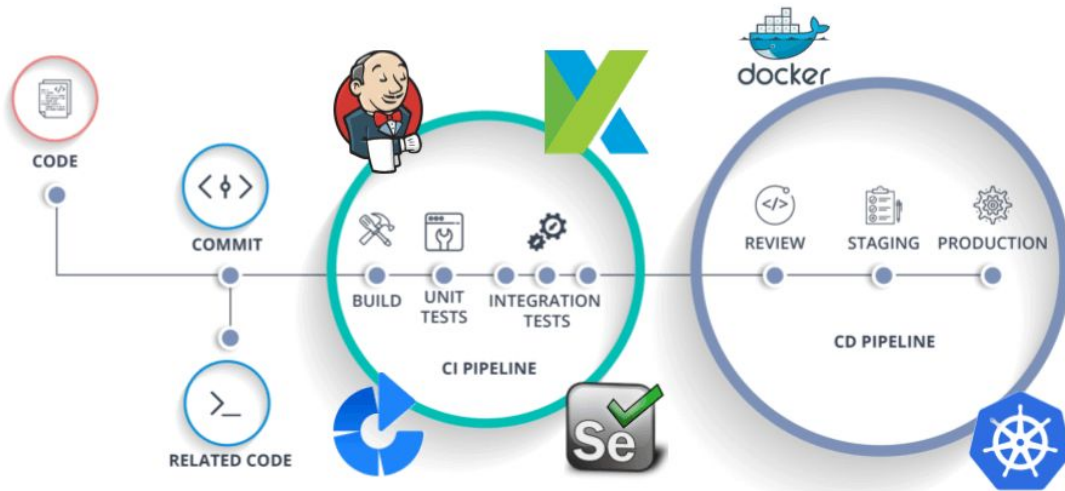
เครื่องมือ DevOps

1. Gradle
2. Git
3. Jenkins
4. Bamboo
5. Docker
6. Kubernetes
7. Puppet Enterprise
8. Ansible
9. Nagios
10. Raygun

2. CI/CD คืออะไร มีความสัมพันธ์กับ DevOps อย่างไร

Ans.

CI/CD Pipeline



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

CI/CD ย่อมาจากคำว่า Continuous Integration และ Continuous Delivery (Deployment) CI เป็นการรวมโค้ดต่างๆของคนในทีมเข้าด้วยกันและทดสอบด้วย test script เพื่อให้มั่นใจว่าไม่มี bug หรือ error ในส่วนต่างของโปรแกรมแล้วจึง commit ไปที่ branch master อีกที โดยจะใช้ตัว build server มาช่วยและเริ่มกระบวนการรวมตั้งแต่เมื่อมีการเปลี่ยนแปลง source code ที่ repository กลาง ระบบจะตรวจสอบโค้ดหลังจากที่เปลี่ยนแปลงว่าทำงานร่วมกันได้หรือไม่หรือเรียกง่าย ๆ คือตัวที่มาช่วยทดสอบโค้ดเพื่อหาว่ามี error ตรงไหนแล้วเอาไปใช้งานต่อ ซึ่งช่วยลดภาระของทีมตรวจสอบนั่นเอง ส่วน CD มีทั้ง Continuous Deployment ที่เป็นการ deploy ขึ้น ซึ่งจะทำการทุกกระบวนการเช่น compile build จนถึง deploy ขึ้น product แบบอัตโนมัติ และ CD อีกอย่างคือ Continuous Delivery ที่จะทำให้ทุกกระบวนการเหมือน CD ตัวแรก แต่ต่างที่จะไม่มีการ deploy แบบอัตโนมัติแต่จะเป็นแบบ manual deploy ซึ่งทั้งสองนี้เป็นเทคนิคในกระบวนการ DevOps หรือที่เรียกว่า pipeline ที่ช่วยให้งานนั้นมีความเร็วยิ่งขึ้น มีเสถียรภาพมากขึ้นเพิ่มความสะดวก และส่งผลให้ Product ที่ Deliver ออกมามีคุณภาพมากขึ้น

3. หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์ จะต้องทำอย่างไรบ้าง

Ans.

นำเอากระบวนการเหล่านี้มาทำหามีลดความหลากหลายของการทำงานที่มีขั้นตอนต่างๆ จัดการกับ Source code ให้ดีเช่นพวก build deploy และทำ Automated test เป็นต้น เช่นกระบวนการทดสอบอัตโนมัติคือจะไปใช้ทดสอบช่วงของกระบวนการ CI ที่ทุกๆการเปลี่ยนแปลงคือการทดสอบแบบอัตโนมัติ ซึ่งอย่างเรื่องพัฒนาของ Developer จะแบ่ง environment เป็น 4 ส่วนคือ local, alpha, staging และ production มีกระบวนการดังนี้

1. Developer เมื่อพัฒนาคุณลักษณะเสร็จจะ สร้าง ทดสอบและรันบนเครื่องตนเองเพื่อทดสอบว่าระบบทำงานได้ถูกและสิ่งที่เปลี่ยนจะไม่กระทบส่วนอื่น

2. ดึงsource code ล่าสุดจาก repository ของระบบเพื่อตรวจสอบว่ามีการเปลี่ยนแปลงหรือไม่ ถ้าเปลี่ยนก็ทำอีกรอบที่ local แล้วส่งไปส่วนกลาง
3. เมื่อส่วนกลางเปลี่ยน จะต้องมียระบบ CI ทำการ build หลังจากนั้นจะส่งไป run unit testing ให้ผ่านทั้งหมดแล้วส่งไปยัง CD เพื่อ deploy
4. เมื่อ source code ถูก deploy ถึง alpha environment แล้วจะ trigger สั่ง run job automated testing ใน level ของ test case ของ smoke test ในส่วนคุณลักษณะที่ถูกdeploy มา
5. หลังจาก run smoke test เสร็จแต่ไม่ผ่านก็จะวนที่เดิมอีกรอบ และจะทำ investigate ว่าเกิดจากอะไร และแก้ไข และ deploy ใหม่เรื่อยๆ
6. ถ้าผ่านทั้งหมด จะส่งต่อไปยังระบบ continuous delivery เพื่อ deploy to Staging environment แล้วก็เช็คทุกระบวนการทำซ้ำๆ จนผ่านทั้งหมดแล้ว business บอกว่าผ่าน แล้วค่อยเอาขึ้น production ได้เพื่อยืนยันเอาโค้ดตัวนี้ไปที่ production environment

อ้างอิง

การพัฒนาระบบตามแนวทาง CI/CD และ DevOps คืออะไร?. [ม.ป.ป.]. **การพัฒนาระบบตามแนวทาง**

CI/CD และ DevOps คืออะไร?. ค้นเมื่อ 24 กุมภาพันธ์ 2564, จาก

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

เกรียงไกร เชาว์นิธิ. (2562). **DevOps คืออะไร นำมาประโยชน์ได้ยังไง และตัวอย่างการทำ DevOps ที่ Credit OK.** ค้นเมื่อ 23 กุมภาพันธ์ 2564, จาก <https://www.spicydog.org/blog/introduction-to-devops-and-the-practical-use-cases-at-credit-ok/>

เทคสตาร์. [ม.ป.ป.]. **รู้จัก “DevOps” ให้มากขึ้น “ตำแหน่ง” ที่องค์กรต่างมองหา.** ค้นเมื่อ 23 กุมภาพันธ์ 2564, จาก <https://www.techstarthailand.com/blog/detail/why-DevOps-skills-continue-to-be-in-high-demand/127>

นริศสราร. (2560). **การนำ Automated testing มารวมเข้ากับ กระบวนการ CI/CD.** ค้นเมื่อ 25 กุมภาพันธ์ 2564, <https://medium.com/@narissara.tan/การนำ-automated-testing-มารวมเข้ากับ-กระบวนการ-ci-cd-58c97e31e230>

บอร์นทูเดฟ ทีม. (2563). **DevOps คืออียังวะ ?**. ค้นเมื่อ 23 กุมภาพันธ์ 2564, จาก <https://www.borntodev.com/2020/05/15/devops-คืออียังวะ/>

ปวีรรต ศักดินิมิตวงศ์. (2561). **Learn DevOps ตอนที่ 2 : DevOps คืออะไร ?**. ค้นเมื่อ 23 กุมภาพันธ์ 2564, จาก https://medium.com/@pariwat_s/learn-devops-ตอนที่-2-devops-คืออะไร-18ac48d73625

สมเกียรติ ปุ้ยสูงเนิน. (2563). **แนวทางในการนำ DevOps มาใช้ปรับปรุงการพัฒนา software.** ค้นเมื่อ 24 กุมภาพันธ์ 2564, จาก <https://www.somkiat.cc/think-before-start-devops/>

แอมะซอน เว็บ เซิร์ฟเวอร์. [ม.ป.ป.]. **DevOps คืออะไร.** ค้นเมื่อ 23 กุมภาพันธ์ 2564, จาก https://aws.amazon.com/th/devops/what-is-devops/?nc1=f_cc