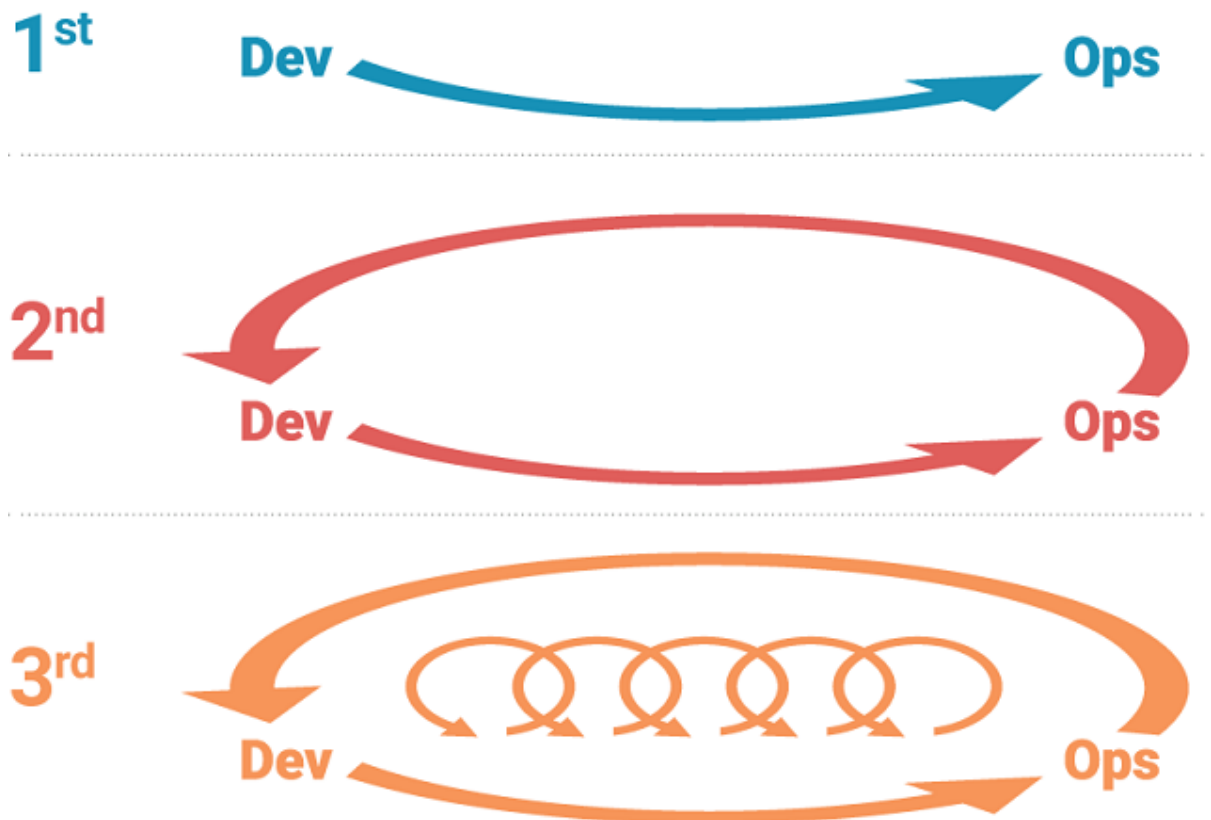


1. DevOps คืออะไร มีความสัมพันธ์กับการพัฒนาซอฟต์แวร์อย่างไร

DevOps เป็นรูปแบบกระบวนการเพื่อแก้ไขปัญหที่เกิดจากฝ่าย Development ที่มีหน้าที่ในการสร้าง feature ให้ทันสมัย ตอบโจทย์ลูกค้า และแข่งขันภายในตลาดได้ และฝ่าย Operations ที่ต้องดูแลเรื่องความปลอดภัยในระบบ โดยในทางธุรกิจจะเน้นเร็วเป็นหลัก และทำให้เกิดปัญหาค้างคาจากระบบตามมาและสะสมได้

การใช้ DevOps จะช่วยเพิ่มประสิทธิภาพในการทำงาน ผลิต software ได้รัวยิ่งขึ้น และมีคุณภาพ ตรงข้ามกับปัญหาหรือ cost ที่ตามมาก็จะลดน้อยลง

3 หลักการสำคัญของ DevOps



1st หลักการของ flow => การปรับปรุง flow การทำงานหรือการส่งต่องานระหว่างหน่วยงาน จากซ้ายไปขวาตั้งแต่ Business เริ่มคิด requirement ไป Development ไป Operations จนถึงลูกค้าให้ไหลไปอย่างราบรื่น และรวดเร็วที่สุด โดยลดขนาดของงานชิ้นใหญ่ เป็นชิ้นย่อย ๆ แล้ว deploy ให้อย่างค่อยเป็นค่อยไป

2nd หลักการของ feedback => หลักการนี้กล่าวถึง feedback ในแง่ของปัญหาระหว่างหน่วยงานจากขามาซ้ายของทุกจุดในกระบวนการทำงานเช่นจาก Operations ไป Development ต้องค้นหาและแก้ไขให้ได้อย่างรวดเร็ว รวมถึงการป้องกันปัญหาไม่ให้เกิดขึ้นอีกเป็นครั้งที่สอง

3rd หลักการของการทดลองและเรียนรู้อย่างต่อเนื่อง => เป็นหลักการเกี่ยวกับการสร้างวัฒนธรรมเพื่อสนับสนุนสองอย่าง อย่างแรกคือการทดลองสิ่งใหม่ๆเพื่อปรับปรุงกระบวนการทำงานซึ่งทำให้เกิดการเรียนรู้จากทั้งความสำเร็จและความล้มเหลว อย่างที่สองคือความเข้าใจว่าการปฏิบัติอย่างต่อเนื่อง เป็นหนทางสู่ความเชี่ยวชาญและความสำเร็จ การทดลองนำสิ่งใหม่ๆมาใช้เพื่อปรับปรุงการทำงานแบบเดิม ๆ จะมีประสิทธิภาพเพิ่มขึ้นเมื่อนำไปใช้กับหลักการของ feedback

2. CI/CD คืออะไร มีความสัมพันธ์กับ DevOps อย่างไร

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

Continuous Deployment (CD) หมายถึง ต้องมีการ Deploy ขึ้น production ทำทุกขั้นตอนตั้งแต่ compile build ไปจนถึง deploy ขึ้น production แบบอัตโนมัติทั้งหมด

หลักการ CI/CD มีความสัมพันธ์ในเชิงนำมาใช้สำหรับการแก้ปัญหาของ DevOps โดยทำงานตั้งแต่การ Plan, Code, Build, Test, Release, Deploy, Operate, Monitor หรือบางที่เรียกสั้นๆ ว่า Pipeline

3. หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์จะต้องทำอย่างไรบ้าง

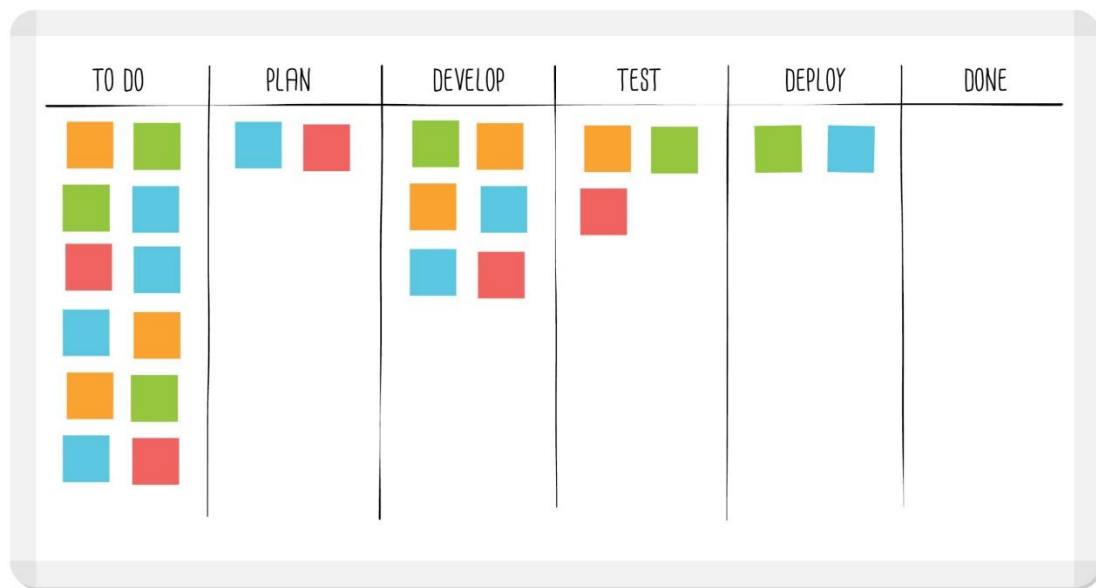
ตัวอย่างขั้นตอนการพัฒนาระบบตามแนวทาง CI/CD

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

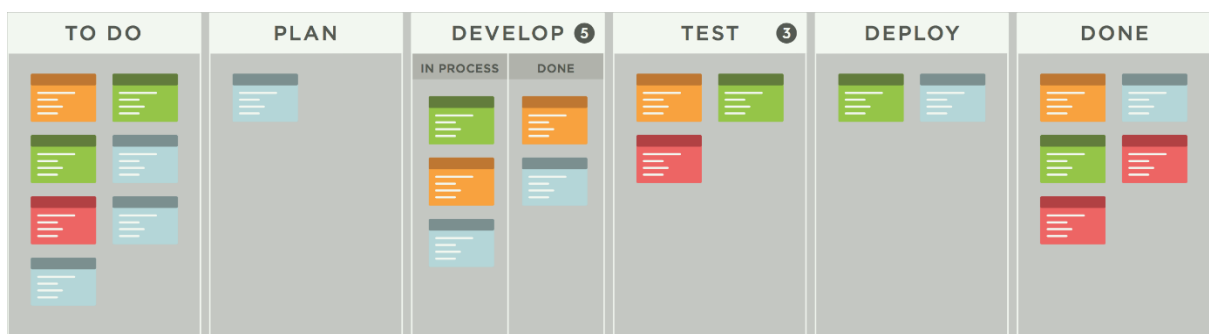
2. ทำการดึง source code ล่าสุดจาก Repository ของระบบ เพื่อตรวจสอบว่ามีการเปลี่ยนแปลงหรือไม่ถ้ามีการเปลี่ยนแปลงก็ให้ทำการรวมหรือ merge ที่เครื่องของ Developer ก่อน จากนั้นจึงทำการ build, test และ run อีกรอบ เมื่อทุกอย่างผ่านทั้งหมด ให้ทำการส่งการเปลี่ยนแปลงไปยัง Repository กลาง
3. เมื่อ Repository กลางมีการเปลี่ยนแปลง จะต้องมีการ 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 ซึ่งเป็นชุดทดสอบย่อย ๆ 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) แล้วจะ trigger ไปสั่งให้ run job automated testing ใน level ของทดสอบ regression test และ QA ก็ทำการทดสอบ Acceptance testing ไปด้วยพร้อมๆ กันที่ UAT(staging) environment นี้ เมื่อมีการ deploy ขึ้นมา เพื่อ fix bug จากที่ QA เจอ หรือที่พบเจอจากการ run regression test แล้ว fail ก็จะเป็นการวน loop ตั้งแต่ต้นจนจบ จนกระทั่งทุกอย่างผ่านหมด Business ยืนยันมาว่าเอาขึ้น production ได้ เป็นการ confirm ว่าเราจะเอา code version สุดท้ายนี้ขึ้นไป production environment

ตัวอย่าง DevOps : หลักการของ Flow

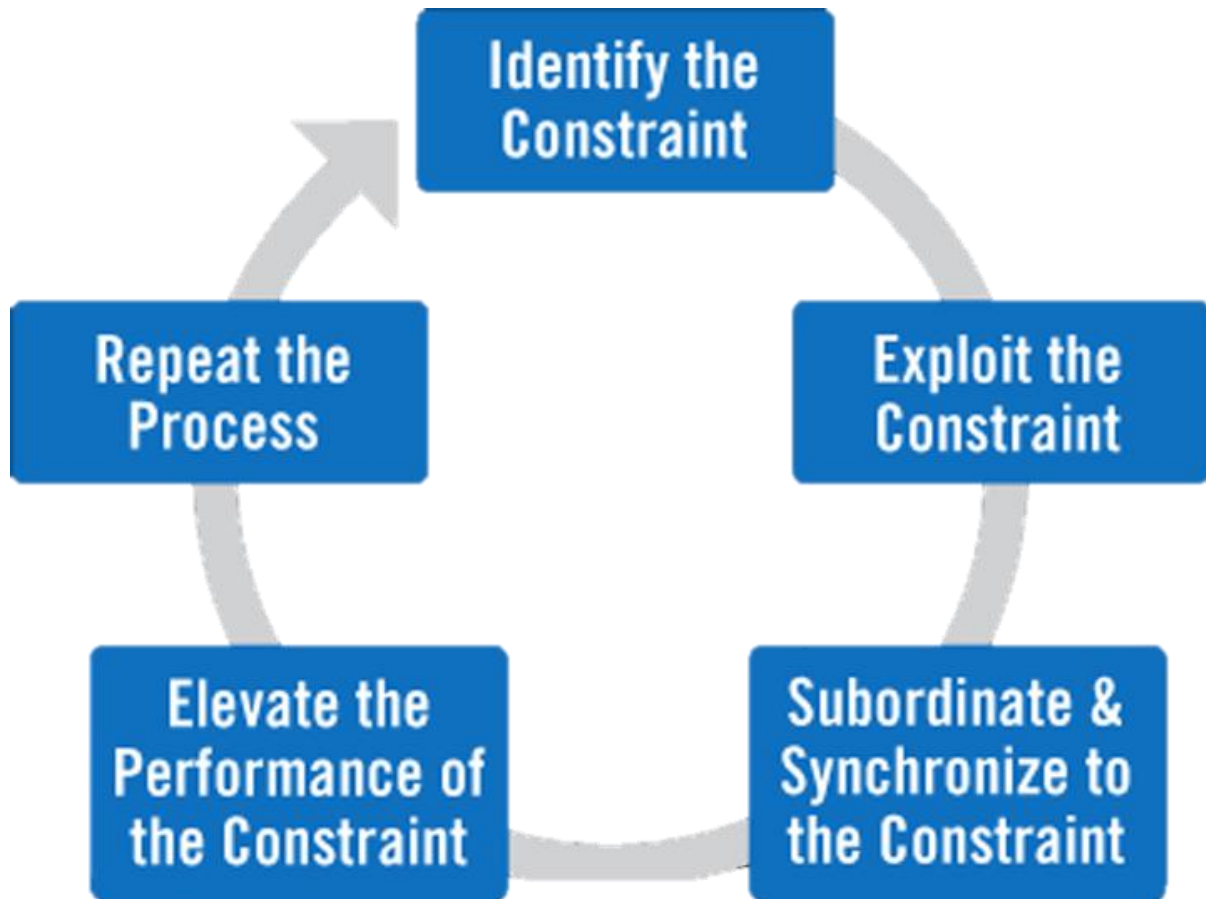
- แสดงภาพของงานทั้งหมดให้ทั้งทีมเห็นอย่างชัดเจน เช่น
 - Kanban Board ให้เห็น flow การทำงานทั้งหมดว่ามีงานอะไรบ้าง อยู่ในคิวกี่งาน มีงานติดขัดอยู่ที่จุดไหน และควรจะทำลำดับความสำคัญของงานอย่างไร



- การจำกัด Work in process
 - จำกัดการ multitasking ได้ด้วยการจำกัด work in process ใน Kanban Board โดย work in process หรือ WIP หมายถึง งานในระหว่างกระบวนการ หรือถ้ามองจาก Kanban Board ก็คืองานที่ไม่ได้อยู่ในคอลัมน์ Done นั่นเอง
 - วิธีการจำกัด WIP ก็คือจำกัดจำนวนการ์ดที่อยู่ในคอลัมน์เหมือนตัวอย่างดังรูป



- การลด Batch size
 - คือการลด Batch size ซึ่งก็คือ ปริมาณการผลิตต่อรอบ
- การลดจำนวนการส่งต่องาน
- การหาจุดคอขวดของ flow การทำงานแล้วกำจัดออกไปอย่างต่อเนื่อง



- อาศัยทฤษฎีข้อจำกัด (Theory of Constraint) ซึ่งเป็นวิธีการเพิ่มอัตราเร็วของทั้งกระบวนการโดยการหาจุดคอขวดแล้วกำจัดออกไปมาช่วย
- การกำจัดความสูญเปล่าใน flow การทำงาน
 - เป็นหลักการที่เน้นลูกค้าเป็นศูนย์กลางโดยปรับปรุงกระบวนการทำงานอย่างต่อเนื่องด้วยการกำจัดความสูญเปล่าที่ไม่ได้เพิ่มคุณค่าให้กับลูกค้า เช่น การรอคอย การสร้าง feature ที่ลูกค้าไม่ได้ใช้งาน เป็นต้น

เอกสารอ้างอิง

Administrator. การพัฒนาระบบตามแนวทาง CI/CD และ DevOps คืออะไร? [Internet]. [cited 2021 Feb 25]. Available from: <https://www.softmelt.com/article.php?id=664>

Junthep T. เมื่อหัดทำ CI/CD ครั้งแรก [Internet]. Medium. 2018 [cited 2021 Feb 25]. Available from: <https://medium.com/@thraithep.a/%E0%B9%80%E0%B8%A1%E0%B8%B7%E0%B9%88%E0%B8%AD%E0%B8%AB%E0%B8%B1%E0%B8%94%E0%B8%97%E0%B8%B3-ci-cd-%E0%B8%84%E0%B8%A3%E0%B8%B1%E0%B9%89%E0%B8%87%E0%B9%81%E0%B8%A3%E0%B8%81-f4f79151311c>

Saknimitwong P. Learn DevOps ตอนที่ 2 : DevOps คืออะไร ? [Internet]. Medium. 2018a [cited 2021 Feb 25]. Available from: https://medium.com/@pariwat_s/learn-devops-%E0%B8%95%E0%B8%AD%E0%B8%99%E0%B8%97%E0%B8%B5%E0%B9%88-2-devops-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-18ac48d73625

Saknimitwong P. Learn DevOps ตอนที่ 3: หลักการของ Flow [Internet]. Medium. 2018b [cited 2021 Feb 25]. Available from: https://medium.com/@pariwat_s/learn-devops-%E0%B8%95%E0%B8%AD%E0%B8%99%E0%B8%97%E0%B8%B5%E0%B9%88-3-%E0%B8%AB%E0%B8%A5%E0%B8%B1%E0%B8%81%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%82%E0%B8%AD%E0%B8%87-flow-167d6f46603c