

#### 11. โปรแกรม version control มีประโยชน์อย่างไร

- ป้องกันความเสียหายที่อาจจะเกิดขึ้นโดยไม่คาดคิด เช่น ไฟดับแล้วลืม save ข้อมูลไว้
  - สามารถเขียน Commit message ได้ ทำให้เราไม่ลืมในขั้นตอนที่เราได้ทำมาก่อนหน้านี้ และสามารถให้คนที่จะมาทำโปรแกรมต่อนั้นมีความเข้าใจในงานที่เรา ทำการ Commit ไว้ได้ด้วย
  - สามารถ ตัดแบ่งส่วนงานเป็นชิ้นย่อย เพื่อให้ระบบทั้งระบบ พัฒนาไปพร้อมๆกันและเสร็จพร้อมกันได้อีกด้วย
  - นอกจากนี้ยังมีโหมด Code reviews ที่จะให้ ผู้อื่นสามารถเห็นโค้ดของเราได้ บางข้อผิดพลาดเราอาจมองเห็นได้ไม่ชัดเจนนัก จึงต้องมีผู้พัฒนาคนอื่นเข้ามาช่วยตรวจสอบ
- 

12. ข้อได้เปรียบ Distributed Version Control Systems (DVCSs) หรือระบบ VCS แบบกระจายศูนย์ ในระบบแบบนี้ คือแต่ละคนไม่เพียงได้ก๊อปปี้ล่าสุดของไฟล์เท่านั้น แต่ได้ทั้งก๊อปปี้ของ repository เลย หมายความว่าถึงแม้ว่าเซิร์ฟเวอร์จะเสีย client ก็ยังสามารถทำงานร่วมกันได้ต่อไป และ repository เหล่านี้ของ client ยังสามารถถูกก๊อปปี้กลับไปเซิร์ฟเวอร์เพื่ออัปเดตข้อมูลกลับคืนก็ได้ การ checkout แต่ละครั้งคือการทำสำเนาข้อมูลทั้งหมดแบบเต็ม ๆ นั่นเอง ซึ่งต่างจาก Centralized Version Control Systems (CVCSs) หรือระบบ Version Control Systems แบบรวมศูนย์ ระบบที่มีการรวมงานไว้ที่ศูนย์กลางเครื่องเดียว ไม่มีการสำรอง ไว้ได้ หากพังจะไม่มีใครสามารถทำงานต่อได้นอกจากต้องใช้เวลากู้ข้อมูลขึ้นมาใหม่เท่านั้น

---

13. ข้อได้เปรียบ Centralized Version Control Systems (CVCSs) หรือระบบ Version Control การจัดการแบบรวมศูนย์ในที่เดียวทำได้ง่ายกว่าการจัดการฐานข้อมูลใน client แต่ละเครื่องเยอะ เพื่อไม่ให้เกิดความวุ่นวายระหว่าง client หลายๆเครื่อง

---

#### 14. แนวทางในการแก้ไขความขัดแย้ง หรือ conflict ที่เกิดขึ้น คือ

- การแบ่งงานและให้แต่ละส่วน merge ขึ้นไป โดยใช้ version control เป็นตัวควบคุม เพื่อให้ได้ซึ่งงานที่มีคุณภาพและไม่เกิดความขัดแย้งในการ merge
-

15. แนวทางในการแก้ไขความขัด หรือ conflict ที่เกิดขึ้น คือ

- อาจจะมีการนัดประชุมและนัดหมายกำหนดการส่งงานในแต่ละส่วนเพื่อที่เวลา merge จะได้ไม่เกิดการขัดแย้งกัน

---

16. Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจกต์เรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชันต่างๆของโปรเจกต์ที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ ฉะนั้น Version Control ก็เหมาะสมอย่างยิ่งสำหรับนักพัฒนาไม่ว่าจะเป็นคนเดียวโดยเฉพาะอย่างยิ่งจะมีประสิทธิภาพมากหากเป็นการพัฒนาเป็นทีม ซึ่งต่างจาก GitHub เพราะ GitHub ก็คือผู้ให้บริการ hosting สำหรับ git เท่านั้น

---

17.จุดประสงค์หลักในการ branch คือ เพื่อแสดงสถานะปัจจุบันของ code ที่อยู่บนระบบ production

---

18. Fast Forward merge คือ การวางแผนระบบให้มีการทำงานที่เดินหน้าไปอย่างรวดเร็ว และเป็นเส้นตรงสวยงาม

ทำไมการ push ไปที่ remote repo จึงควรจะต้องใช้ merge แบบนี้ก็เพราะว่า วิธีนี้จะสลับ Head ไปทำให้มันไม่เกิดปัญหาและไม่มีการ Conflict กัน

---

19.หน้าที่หลักของ git pull คือ ดึงสิ่งใหม่ๆจาก origin ลงมา merge ทั้งบน clone repository และ working directory โดยทันที หากเป็นมี conflict จากการ merge ใน working directory เราต้อง resolve conflict นั้นๆก่อนจะ commit ได้ต่อไป

---

20. สื่อความหมายถึง ทำ release branch จะเรียกว่า Branch by release นั่นเองในแต่ละ branch คือ แยกไปตามแต่ละ version ของระบบงานแต่ส่วนการทำงานหลักยังคงอยู่ที่ branch หลัก

---