

11. โปรแกรม Version control มีประโยชน์อย่างไร?

**ตอบ** ใช้เป็นจัดการไฟล์งาน ในกรณีที่เรารู้หลายเวอร์ชัน หากเราทำงานเสียหายหรือต้องการสิ่งบางอย่างที่อยู่ในไฟล์เวอร์ชันเก่า ก็สามารถเรียกนำมาใช้งานได้ผ่าน Version Control

อีกทั้งยังสามารถดูประวัติการแก้ไขไฟล์ได้ด้วย (หากใช้งานหลายคน) Version Control System จะมีระบบ log จัดการ ควบคุม เปรียบเทียบไฟล์ระหว่างเวอร์ชันก่อนหน้านี้กับเวอร์ชันปัจจุบัน อีกทั้ง หากคุณหรือคนในทีมเผลอลบไฟล์ Version Control system ก็สามารถกู้คืนไฟล์ที่เผลอลบไปได้...

12. ข้อได้เปรียบของ distributed version control เมื่อเทียบกับ centralized version control คืออะไร

**ตอบ** เหมาะกับการทำงานเป็นทีม เนื่องจาก client ที่เข้ามาใช้งาน สามารถก๊อปปี้ทุกอย่างทั้งหมดบน repo (ไฟล์, ประวัติการแก้ไข ฯลฯ) ไปเก็บบันทึกไว้ในเครื่องได้ ซึ่งถึงแม้ว่าเซิร์ฟเวอร์จะเสีย แต่ client ก็ยังทำงานได้ตามปกติ อีกทั้ง หากมีการลบไฟล์หรือไฟล์หายบนเซิร์ฟเวอร์ ตัว client ก็สามารถอัปเดตข้อมูลกลับขึ้นไปได้ทั้งหมด เสมือน client ทำการแบ็คอัพให้ตัวเซิร์ฟเวอร์ repo ในกรณีที่เกิดเหตุไม่คาดฝัน งานเราก็จะไม่หาย และนี่คือข้อได้เปรียบของ distributed version control

13. ข้อได้เปรียบของ centralized version control เมื่อเทียบกับ distributed version control คืออะไร

**ตอบ** ประหยัดพื้นที่ในเครื่อง local เนื่องจาก centralized version control ไม่จำเป็นต้องก๊อปปี้ทุกอย่างบน repo มาลงเครื่อง ผิดกับ distributed version control ที่ต้องก๊อปปี้ทุกอย่างลงมาเก็บใน local ถ้าหากไฟล์บน repo มีขนาดใหญ่ และมีการบันทึก log การแก้ไขมาก ( $\geq 50,000$  lines) คงไม่สนุกนักกับการรอดาวน์โหลดทุกอย่างลงจาก repo และนี่คือข้อได้เปรียบของ centralized version control

14. บอกแนวทางในการแก้ไข conflict ที่เกิดขึ้นเมื่อมีการ merge โปรแกรมของผู้พัฒนาหลายคนเข้าด้วยกัน

**ตอบ** สั่ง pull ลงเครื่อง แล้วเปิดไฟล์เจ้าปัญหาขึ้นมา แก้ไขโค้ดส่วนเกินออกซะ แล้วก็สั่ง commit & push ใหม่ ปัญหา ก็จะหมดไป

15. บอกแนวทางในการลด conflict ที่จะเกิดขึ้นจากการ merge โปรแกรมของผู้พัฒนาหลายๆคนเข้าด้วยกัน

**ตอบ**

1. Merge บ่อยๆ อย่าอดงัดไฉนนาน
2. พูดคุย สื่อสารกันเยอะๆ บอกกันดีๆ

16. Git คืออะไร แตกต่างจาก Github อย่างไร

**ตอบ** Git คือ Version Control System ตัวหนึ่ง ที่มีความสามารถหลากหลายดังที่รู้จักกัน ส่วน Github คือ Service ที่ให้บริการพื้นที่เก็บ Git repo ของเรา เพื่อให้สามารถใช้งานบนโลกอินเทอร์เน็ตได้

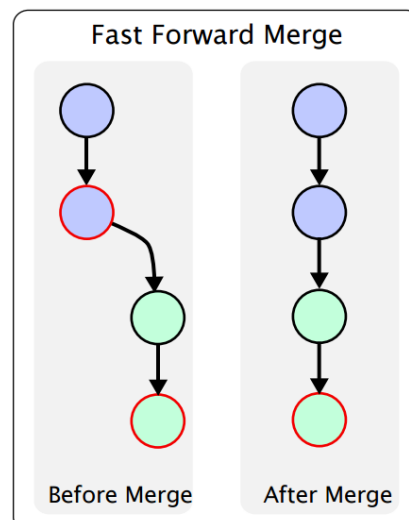
17. จุดประสงค์หลักในการทำ branch คืออะไร

**ตอบ** แยกออกมาเพื่อให้ทำงานสะดวกขึ้น ในกรณีที่ไค้ด้นั้นๆ stable แล้ว แต่อยากจะทำอะไรเพิ่มเติมกับมัน โดยขณะเดียวกัน เราก็ไม่อยากจะไปแตะต้องตัว repo หลัก กลัวจะเกิดปัญหา การทำ branch หรือ sub repo จึงช่วยแก้ปัญหาในเรื่องนี้

ซึ่งเมื่อเสร็จงานแล้ว เราสามารถสั่ง merge รวมเข้ากับ repo ตัวหลักที่เราแยกออกมาได้

18. Fast forward merge คืออะไร และทำไมการ push ไปที่ remote repo จึงควรจะต้อง merge แบบนี้

**ตอบ** เป็นการบังคับให้ประวัติเส้นทางวิ่งเป็นเส้นตรง (ดูรูปประกอบด้านล่าง) และที่นำไปที่ remote repo เพื่อป้องกันความเสียหายจากการ conflict ที่อาจตามมา



19. หน้าที่หลักของคำสั่ง git pull คืออะไร

**ตอบ** ดึงโค้ดจาก remote มายัง local

20. แผนภาพด้านล่างนี้ต้องการสื่อความหมายอะไร

**ตอบ** เป็นตัวอย่างการทำ branch แยกแขนงออกไป ก่อนที่จะนำไป merge รวมในตัว master