

11. **ตอบ** ช่วยให้เราสามารถย้อนไฟล์บางไฟล์หรือแม้กระทั่งทั้งโปรเจกต์กลับไปเป็นเวอร์ชันเก่าได้ นอกจากนั้นระบบ Version Control ยังจะช่วยให้สามารถเปรียบเทียบการแก้ไขที่เกิดขึ้นในอดีต ดูว่าใครเป็นคนแก้ไขคนสุดท้ายที่อาจทำให้เกิดปัญหา แก้ไขเมื่อไร และยังช่วยให้สามารถกู้คืนไฟล์ที่ลบไปหรือทำเสียโดยไม่ตั้งใจได้อย่างง่ายดาย

12. **ตอบ** สามารถ commit ได้แบบ offline โดยไม่ต้องผ่าน Network จึงเร็วกว่า และผู้ใช้งานก็สามารถแก้ไขได้ทุกที่โดยไม่จำเป็นต้องติดต่อกับ central repo ก็ได้ และมันช่วยกระจายความเสี่ยงของ Project History จะหาย เพราะพังไปเครื่องหนึ่งเครื่องอื่นๆ ก็ยังมีอยู่ไม่หายไปไหน

13. **ตอบ** เมื่อเริ่มต้นเราก็สร้างโปรเจกต์ขึ้นมาบน Server แต่ละคนก็สามารถมาดาวน์โหลดเครื่องของตัวเองได้ เวลาที่มีการเปลี่ยนแปลงอะไรก็อัปโหลดกลับขึ้นไปบน Server แล้วคนอื่นๆ ก็จะสามารถดาวน์โหลด Update นั้นมายังเครื่องของตัวเองได้

14. **ตอบ** เมื่อทำการ git merge จะรวมโค้ดให้เราเองอัตโนมัติ แต่ก็จะมีข้อยกเว้นเมื่อแก้ไขไฟล์เดียวกัน ในกรณีที่เราและเพื่อนร่วมทีม แก้ไขไฟล์เดียวกัน Git จะเกิดการ conflict เมื่อเราจะ merge โค้ด โดยไม่รู้ว่าจะใช้โค้ดของเราหรือของเพื่อน วิธีแก้ก็คือ ทำการ edit แล้ว commit ไปใหม่

15. **ตอบ** ตกลงกับเพื่อนในทีมให้ดีกว่ากันว่าใครจะทำงานไหนเพื่อที่จะไม่ทำซ้ำไฟล์เดียวกัน

16. **ตอบ**

- ☐ Github เป็นเว็บเซิร์ฟเวอร์ที่ให้บริการในการฝากไฟล์ Git (ทั่วโลกมักนิยมใช้ในการเก็บโปรเจกต์ Open Source ต่างๆ)
- ☐ Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจกต์เรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชันต่างๆของโปรเจกต์ที่ใดเวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ ฉะนั้น Version Control ก็เหมาะสมอย่างยิ่งสำหรับนักพัฒนาไม่ว่าจะเป็นคนเดียว โดยเฉพาะอย่างยิ่งจะมีประสิทธิภาพมากหากเป็นการพัฒนาเป็นทีม

17. **ตอบ** การที่เรามีโค้ดที่คีย์อยู่แล้ว แต่อยากจะทดลองอะไรนิดๆหน่อยๆ หรือแก้ไขอะไรก็ตาม ไม่ให้กระทบกับตัวงานหลัก ก็เพียงแค่สร้าง branch ใหม่ขึ้นมา เมื่อแก้ไขหรือทำอะไรเสร็จแล้ว ก็ค่อยเชฟกลับมาที่ master เหมือนเดิม

18. **ตอบ** การทำ fast forward จะเป็นการที่ commit ที่ถูกชี้โดย branch ที่เราทำการ merge มันเป็น upstream ของ commit ที่เราอยู่โดยตรง Git ก็เลยขยับ pointer ไปข้างหน้า หมายถึง เวลาที่เราพยายามจะ merge commit ซักอันเข้ากับ commit ที่สามารถไปถึงได้โดยการตาม history ของ commit อันแรก Git จะทำให้ทุกอย่างง่ายขึ้นโดยการขยับ pointer ไปข้างหน้าเพราะมันไม่มีงานที่ถูกแยกออกไปให้ merge

19. **ตอบ** การรวมโค้ดจาก remote มายัง local โดยที่เราไม่สามารถรู้ได้เลยว่าจะรวมโค้ดอะไรบ้าง จะรู้เมื่อรวม pull เสร็จแล้ว ดังนั้น git pull ก็คือการทำ git fetch และต่อด้วย git merge โดยอัตโนมัติ

20. **ตอบ** การทำโปรเจกต์และมีการพัฒนาแก้ไข แล้วอัปเดตงานมารวมกัน และสามารถนำงานไปพัฒนาแก้ไขต่อได้อีก