**11. โปรแกรม version control มีประโยชน์อย่างไร**

- การใช้ Version Control System (VCS) มีประโยชน์มันช่วยให้คุณสามารถย้อนไฟล์บางไฟล์หรือแม้กระทั่งทั้งโปรเจคกลับไปเป็นเวอร์ชั่นเก่าได้ นอกจากนั้นระบบ VCS ยังจะช่วยให้คุณเปรียบเทียบการแก้ไขที่เกิดขึ้นในอดีต ดูว่าใครเป็นคนแก้ไขคนสุดท้ายที่อาจทำให้เกิดปัญหา แก้ไขเมื่อไร ฯลฯ และยังช่วยให้คุณสามารถกู้คืนไฟล์ที่คุณลบหรือทำเสียโดยไม่ตั้งใจได้อย่างง่ายดาย

**12. ข้อได้เปรียบของ distributed version control เมื่อเทียบกับ centralized version control คืออะไร**

- ถึงแม้ว่าเซิร์ฟเวอร์จะเสีย client ก็ยังสามารถทำงานร่วมกันได้ต่อไป และ repository เหล่านี้ของ client ยังสามารถถูกก๊อปปี้กลับไปที่เซิร์ฟเวอร์เพื่อกูข้อมูลกลับคืนก็ได้ การ checkout แต่ละครั้งคือการทำสำรองข้อมูลทั้งหมดแบบเต็ม ๆ นั่นเอง

**13.ข้อได้เปรียบของ centralized version control เมื่อเทียบกับ distributed version control คืออะไร**

- ทุกคนสามารถรู้ได้ว่าคนอื่นในโปรเจคกำลังทำอะไร ผู้ควบคุมระบบสามารถควบคุมได้อย่างละเอียดว่าใครสามารถแก้ไขอะไรได้บ้าง การจัดการแบบรวมศูนย์ในที่เดียวทำได้ง่ายกว่าการจัดการฐานข้อมูลใน client แต่ละเครื่องเยอะ

**14. บอกแนวทางการแก้ไข confict ที่เกิดขึ้นเมื่อมีการ merge โปรแกรมของผู้พัฒนาหลายๆ คนเข้าด้วยกัน**

- ทำการ edit แล้ว commit ไปใหม่

15.บอกแนวทางการลด confict ที่เกิดขึ้นเมื่อมีการ merge โปรแกรมของผู้พัฒนาหลายๆ คนเข้าด้วยกัน

- การแยกส่วนการทำงานกันให้เรียบร้อย หากมีคนแก้ไขก็ใช้การเปลี่ยนชื่อไฟล์แล้วค่อย push

**16. git คืออะไร แตกต่างจาก github อย่างไร**

- Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็คเรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของโปรเจ็คที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ ฉะนั้น Version Control ก็เหมาะอย่างยิ่งสำหรับนักพัฒนาไม่ว่าจะเป็นคนเดียวโดยเฉพาะอย่างยิ่งจะมีประสิทธิภาพมากหากเป็นการพัฒนาเป็นทีม

ต่างจาก github เป็นเว็บไซต์ที่ให้บริการเสมือน git server เป็นบริการฟรีแบบมีเงื่อนไข คุณสามารถใช้งานได้ฟรี แต่โปรเจ็กที่สร้างขึ้นจะต้องเป็นแบบ public เท่านั้น คือ คนอื่นสามารถเข้าไปดูโค้ด ดูดโค้ด ท่านได้ ถ้าไม่อยากให้ชาวบ้านเห็นโปรเจ็กของท่าน ก็ต้องเสียตังใช้บริการ เพื่อสร้างโปรเจ็กแบบ private นั่นหมายความว่าคุณต้องเป็นคนใจกว้าง 555 คือเปิดโค๊ดโปรเจ็คให้เพื่อนดู แต่หากเราไม่อยากให้เพื่อดูเราก็ต้องจ่ายตัง อ่าวหลายคนบอกแบบนี้ก็แย่อ่ะดิ ไม่หรอกครับในมุมมองของผม มองว่า github มันเหมือนเป็นแหล่งรวม sourcecode ดีๆนี่เอง

**17. จุดประสงค์การใช้ branch ตืออะไร**

- เป็น feature ที่ช่วยให้นักพัฒนาสามารถที่จะทำงานได้สะดวกขึ้น ยกตัวอย่างเช่น เรามีโค๊ดที่ดีอยู่แล้ว แต่อยากจะทดลองอะไรนิดๆหน่อย หรือแก้ไขอะไรก็ตาม ไม่ให้กระทบกับตัวงานหลัก ก็เพียงแค่สร้าง branch ใหม่ขึ้นมา เมื่อแก้ไขหรือทำอะไรเสร็จแล้ว ก็ค่อยเซฟกลับมาที่ master เหมือนเดิม

**18. Fast forward merge คืออะไรและทำไมการ push ไปที่ remote repo จึงควรจะต้อง merge แบบนี้**

-

**19. หน้าที่หลักของ git pull คืออะไร**

- คือรวมโค๊ดจาก remote มายัง local โดยที่เราไม่สามารถรู้ได้เลยว่าจะรวมโค๊ดอะไรบ้าง รู้แค่หลังจาก pull เสร็จแล้วนั่นเอง สรุป git pull มันก็คือการทำ git fetch และต่อด้วย git merge อัตโนมัตินั่นเอง

**20. แผนภาพด้านล่างต้องก่ีสื่ออะไร**

- Mainline Branch Strategy เป็นวิธีการที่เรียบง่ายที่สุดแถมยังมีประสิทธิภาพอย่างมากสำหรับทีมที่ไม่ใหญ่มากเนื่องจากจะทำงานบน branch หลักเดียวเท่านั้นโดยที่ branch หลักนี้ ต้องพร้อมที่จะ deploy/release อยู่ตลอดเวลา ในการทำงานนั้น developer แต่ละคนจะทำงานบน local branch ของตัวเองเมื่อทำการพัฒนาเสร็จแล้วให้ทำการ merge code เขาไปยัง branch หลัก