

11. โปรแกรม version control มีประโยชน์อย่างไร

ระบบที่จัดการการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์เพื่อให้สามารถเรียกเวอร์ชันใดเวอร์ชันหนึ่งกลับมาเมื่อไรก็ได้

12. ข้อได้เปรียบของ distributed version control เมื่อเทียบกับ centralized version control คืออะไร

แต่ละคนไม่เพียงได้ copy ล่าสุดของไฟล์เท่านั้น แต่ได้ทั้ง copy ของ repository เลย หมายความว่าถึงแม้ว่าเซิร์ฟเวอร์จะเสีย client ก็ยังสามารถทำงานร่วมกันได้ต่อไป และ repository เหล่านี้ของ client ยังสามารถถูก copy กลับไปที่ server เพื่อกู้ข้อมูลกลับคืนก็ได้ การ checkout แต่ละครั้งคือการทำสำเนาข้อมูลทั้งหมดแบบเต็ม ๆ นั้นเอง

13. ข้อได้เปรียบของ centralized version control เมื่อเทียบกับ distributed version control คืออะไร

การทำงานแบบ Centralized Version Control ทุกคนสามารถรู้ได้ว่าคนอื่นในโปรเจกต์กำลังทำอะไร ผู้ควบคุมระบบสามารถควบคุมได้อย่างละเอียดว่าใครสามารถแก้ไขอะไรได้บ้าง การจัดการแบบรวมศูนย์ในที่เดียวทำได้ง่ายกว่าการจัดการฐานข้อมูลใน client แต่ละเครื่องเยอะ

14. บอกแนวทางการแก้ไข conflict ที่เกิดขึ้นเมื่อมีการ merge โปรแกรมของผู้พัฒนาหลายคนเข้าด้วยกัน

โดยปกติแล้ว git merge จะรวม code ให้เราเองอัตโนมัติ แต่ก็จะมีข้อยกเว้นเมื่อแก้ไขไฟล์เดียวกัน Git จะเกิดการ conflict เมื่อเราจะ merge code โดยไม่รู้ว่าจะใช้ code ของใคร วิธีแก้คือ ทำการ edit แล้ว commit ไปใหม่

15. บอกแนวทางในการลด conflict ที่เกิดขึ้นจากการ merge โปรแกรมของผู้พัฒนาหลายคนเข้าด้วยกัน

Mob programming เป็นวิธีการที่ทรงประสิทธิภาพอย่างมากและเชื่อได้เลยว่า แก้ไขปัญหา Merge conflict ได้ 100% เนื่องจากทุกคนมาทำงานด้วยกัน ใช้เครื่องทำงานเดียวกัน ดังนั้น ไม่มีทางที่ source code จะขัดแย้งกัน แต่มันมีค่าใช้จ่ายที่สูงมาก

16. Git คืออะไร แตกต่างจาก Github อย่างไร

Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดการการเปลี่ยนแปลงของไฟล์ในโปรเจกต์เรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชันต่าง ๆ ของโปรเจกต์ที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้น ๆ ใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้น ๆ ถูกเขียนโดยใครบ้างก็สามารถทำได้ ฉะนั้น Version Control ก็เหมาะสมอย่างยิ่งสำหรับนักพัฒนาไม่ว่าจะเป็นคนเดียวโดยเฉพาะอย่างยิ่งจะมีประสิทธิภาพมากหากเป็นการพัฒนาเป็นทีม

Github คือ เว็บไซต์ให้บริการพื้นที่จัดเก็บโครงการโอเพ่นซอร์สด้วยระบบควบคุมเวอร์ชันแบบ Git โดยมีจุดประสงค์หลักคือ ทำให้การแบ่งปันและพัฒนาโครงการต่าง ๆ ด้วยกันเป็นไปได้ง่าย ๆ เป็นเว็บไซต์ที่ให้บริการเสมือน git server เป็นบริการฟรีแบบมีเงื่อนไข คุณสามารถใช้งานได้ฟรี แต่โปรเจกต์ที่สร้างขึ้นจะต้องเป็นแบบ public เท่านั้น

17. จุดประสงค์หลักในการ **branch** คืออะไร

เพื่อให้ทำงานได้สะดวก ยกตัวอย่างเช่น เรามีโค้ดที่ดีอยู่แล้ว แต่อยากจะทดลองสิ่งใหม่ๆ หรือแก้ไขอะไรก็ตาม ไม่ให้กระทบกับตัวงานหลัก ก็เพียงแค่สร้าง **branch** ใหม่ขึ้นมา เมื่อแก้ไขหรือทำอะไรเสร็จแล้ว ก็ค่อย **merge** กลับมาที่ **master** เหมือนเดิม

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

เวลาที่คุณพยายามจะ **merge commit** ชั้กันเข้ากับ **commit** ที่สามารถไปถึงได้โดยการตาม **history** ของ **commit** อันแรก **Git** จะทำให้ทุกอย่างง่ายขึ้นโดยการขยับ **pointer** ไปข้างหน้าเพราะมันไม่มีงานที่ถูกแยกออกไปให้ **merge**

19. หน้าที่หลักของคำสั่ง **git pull** คืออะไร

git pull คือรวม **code** จาก **remote** มายัง **local** โดยที่เราไม่สามารถรู้ได้เลยว่าจะรวมโค้ดอะไรบ้าง ซึ่งจริงๆ แล้ว **git pull** มันก็คือการทำ **git fetch** และต่อด้วย **git merge** อัตโนมัติ

20. ความหมายของแผนภาพคือ

git branch model เป็นกระบวนการแยก **branch** ออกจาก **master** และ **merge** กลับเข้าไปภายหลัง