11. โปรแกรม version control มีประโยชน์ไร

ตอบ เก็บรักษาซอร์สโค๊ด และไฟล์รีซอร์สต่าง ๆ ของโปรแกรมที่เราทำงานด้วย โดยมันจะเก็บการเปลี่ยนแปลงของทุกไฟล์ในระบบเป็น version หรือ revision ของไฟล์ เพื่อที่ว่าเราสามารถดูประวัติย้อนหลังไปได้ว่าโค๊ดของโปรแกรมนั้นเคยถูกเปลี่ยนแปลงอะไรมาบ้าง

12. ข้อได้เปรียบของ distributed version control เมื่อเทียบกับ centralized version control คืออะไร

ตอบ distributed version control เร็วกว่า ยืดหยุ่นมากกว่าและผู้ใช้จะมีอำนาจหน้าที่มากกว่า ถ้าไม่สามารถเชื่อมต่อเครือข่ายกับ Server ได้เราก็ยังสามารถทำงานได้และยังสามารถเก็บรุ่นของไฟล์พร้อมทั้งประวัติการแก้ไขและเวลาไว้โดยที่เราไม่จำเป็นต้องมี Server

13. ข้อได้เปรียบของ centralized version control เมื่อเทียบกับ distributed version control คืออะไร

ตอบ centralized version control นั้นจะดูแลรักษาได้ง่ายกว่า และไม่ซับซ้อน

14. บอกแนวทางในการแก้ไข conflict ที่เกิดขึ้นเมื่อมีการ merge โปรแกรมของผู้พัฒนาหลายๆคนเข้าด้วยกัน

ตอบ เป็นวิธีการที่ทรงประสิทธิภาพอย่างมากและเชื่อได้เลยว่า แก้ไขปัญหา Merge conflict ได้ 100% เนื่องจากทุกคนมานั่งทำงานด้วยกันใช้เครื่องทำงานเดียวกัน ดังนั้น ไม่มีทางที่ source code จะขัดแย้งกัน (แต่มันมีค่าใช้จ่ายที่สูงมาก ๆ)

15. บอกแนวทางในการลด conflict ที่จะเกิดขึ้นจาก merge โปรแกรมของผู้พัฒนาหลายๆคนเข้าด้วยกัน

ตอบ การออกแบบที่ดีมันช่วยให้ทีมทำงานร่วมกันได้อย่างดี ยิ่งแต่ละส่วนการทำงานเล็ก ๆ แล้วก็ยิ่งทำให้คุณภาพของการออกแบบระบบดีรวมทั้ง ลดความเสี่ยงจาก Merge conflict อีกด้วย

16. Git คืออะไร แตกต่างจาก Github อย่างไร

ตอบ Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็คเรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของโปรเจ็คที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ แต่ GitHub คือ เว็บที่ให้บริการพื้นที่จัดเก็บโครงการโอเพ่นซอร์สด้วยระบบควบคุมเวอร์ชันแบบ Git โดยมีจุดประสงค์หลักคือ ทำให้การแบ่งปันและพัฒนาโครงการต่างๆด้วยกันเป็นไปได้ง่ายๆ

17. จุดประสงค์หลักในการ branch คืออะไร

ตอบ ช่วยเก็บรักษา code รุ่นที่ยังทำงานถูกต้อง

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

ตอบ ถ้า Head ของเราอยู่ใน commits ที่ดึงมา, case นี้มักเกิดจากคำสั่ง "git pull" เพื่อดึง code จากต้นน้ำมา update code(ที่ไม่มีการเปลี่ยนแปลง) ของเรา, สิ่งที่เกิดขึ้นก็คือ git จะ update HEAD ของเราให้ตรงตาม HEAD ของ merged commit (โดยไม่มีการสร้าง commit object ใหม่ขึ้นมา) มีศัพท์เฉพาะสำหรับกรณีนี้ว่า "Fast-forward"

19. หน้าที่หลักของคำสั่ง git pull คืออะไร

ตอบ ใช้ดึงความเปลี่ยนแปลงจาก remote มายัง local และรวมเข้าด้วยกัน (มีค่าเท่ากับ fetch + merge)

20. แผนภาพนี้ต้องการสื่อความหมาย Now suppose that we have two feature branches. First one which ends with commit A and second one which ends with commit B.

Suppose that there are some issues with feature B and it goes to next releases. In meantime there are few similar to A branches which ends merged into develop.

Branch develop has evolved because of my feature branches. That's why at some point my branch B can't be merged into develop.