

11. ตอบ

- ☐ สามารถ Backup และ Restore ไฟล์ได้
- ☐ Synchronization ไฟล์ทำให้ไฟล์นั้นเป็นเวอร์ชันใหม่เสมอไม่ว่าจะมีผู้ใช้งานไฟล์นั้นกี่คนก็ตาม
- ☐ สามารถ Track back history ของแต่ละไฟล์ได้
- ☐ ใน History นั้นสามารถบอกรายละเอียดได้มากกว่าการจับแบบธรรมดา คือสามารถบอกได้ว่าไฟล์นี้ใครเป็นเจ้าของ และใครบ้างที่แก้ไขไฟล์นี้
- ☐ ควบคุมจัดการได้ง่ายเมื่อต้องการควบคุมโค้ดที่พัฒนาโดยโปรแกรมเมอร์หลายคนในทีม

12. ตอบ ข้อได้เปรียบของ distributed version control คือการที่แต่ละเว็บไซต์ 'สามารถอยู่รอด' ในตัวเอง ไม่มีจุดเดียวของความล้มเหลวในเรื่องนี้เป็น นอกจากนี้สมมติว่าฮาร์ดแวร์ในบางส่วนของเว็บไซต์ที่จะถูกเก็บไว้ในห้องเซิร์ฟเวอร์ที่ปลอดภัย และไม่ได้มีเครื่องใช้สำนักงาน (สมมติฐานใหญ่ในบางกรณีนี้) นอกจากนี้ยังจะอาจอำนวยความสะดวกในธุรกิจอย่างต่อเนื่องโดยใช้เว็บไซต์ที่อ้างอิงแต่ละอื่น ๆ เป็นเว็บไซต์ฉุกเฉิน

13. ตอบ ข้อได้เปรียบของ centralized version control คือทุนที่ลดลงและค่าใช้จ่ายในการดำเนินงาน (ฮาร์ดแวร์น้อยที่สุดในแต่ละสถานที่เดียวกัน), การรักษาความปลอดภัย (ข้อมูลทั้งหมดที่เก็บไว้ในดาต้าเซ็นเตอร์ที่มีความปลอดภัย) ค่าใช้จ่ายในการบริหารน้อยกว่า (ทรัพยากรน้อยลงจำเป็นตั้งแต่อุปกรณ์ทั้งหมดที่อยู่ในสถานที่หนึ่ง) ความซับซ้อนของการสำรองข้อมูลน้อย การควบคุมและมากขึ้นกว่าพื้นที่ที่มีความเสี่ยงที่อาจเกิดขึ้นเช่นการเข้าถึงอินเทอร์เน็ต

14. ตอบ

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

Single Responsibility Principle (SRP) เป็นหนึ่งในแนวทางการออกแบบระบบงานที่ดี คือ SOLID นั่นคือ ในหนึ่ง class ใน หนึ่ง method นั้นควรจะมีความที่การทำงานเพียงอย่างเดียวเท่านั้น หรือในแต่ละ class แต่ละ method ควรมีเหตุผลเดียวในการเปลี่ยนแปลงเท่านั้น ผลที่ได้ก็คือ นักพัฒนาจะไม่ทำงาน หรือ เปลี่ยนแปลง source code ที่เดียวกันอย่างแน่นอน ยกเว้นจะทำงานเดียวกัน หรือ ทำงานด้วยกัน ดังนั้น การออกแบบที่ดีมันช่วยให้ทีมทำงานร่วมกันได้ดี ยิ่งแต่ละส่วนการทำงานเล็ก ๆ แล้วก็จะทำให้คุณภาพของการออกแบบระบบโดยรวมทั้ง ลดความเสี่ยงจาก Merge conflict อีกด้วย

15. ตอบ สามารถดึงการเปลี่ยนแปลงล่าสุดจาก remote repository มายัง local repository ด้วยคำสั่ง git pull โดย git จะทำการ ดึงข้อมูล และ รวมข้อมูล การเปลี่ยนแปลงจาก remote repository ในเครื่องของคุณ ถ้าต้องการ merge การ

เปลี่ยนแปลงจาก branch อื่นมายัง branch ปัจจุบัน (เช่น master) ด้วยคำสั่ง `git merge <branch>` ซึ่งทั้งสองคำสั่ง ทำการ merge อย่างอัตโนมัติ

16. **ตอบ** Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจกต์เรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชันต่างๆของโปรเจกต์ที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆ ใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ ฉะนั้น Version Control ก็เหมาะอย่างยิ่งสำหรับนักพัฒนาไม่ว่าจะเป็นคนเดียวโดยเฉพาะอย่างยิ่งจะมีประสิทธิภาพมากหากเป็นการพัฒนาเป็นทีม

Github เป็นเว็บไซต์เวอร์ที่ให้บริการในการฝากไฟล์ Git (ทั่วโลกมักนิยมใช้ในการเก็บโปรเจกต์ Open Source ต่างๆ ที่ดังๆ ไม่ว่าจะเป็น Bootstrap, Rails, Node.js, Angular เป็นต้น)

17. **ตอบ** branch คือ การแบ่งหรือแตกกิ่ง โดยเริ่มต้นจากปัญหาที่มีขนาดใหญ่แบ่งเป็นปัญหามาน้อยๆ จากนั้นพิจารณาขอบเขตของคำตอบสำหรับปัญหาย่อย และพิจารณาตัดปัญหาที่ไม่สามารถให้คำตอบได้ดีที่สุด และทำซ้ำกับปัญหาย่อยๆ จนกระทั่งพบปัญหาย่อยที่ให้คำตอบได้ดีที่สุด ซึ่งวิธีนี้จะสามารถหาคำตอบที่ดีที่สุดได้ในเวลาอันรวดเร็ว

18. **ตอบ** Fast forward merge คือ 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. **ตอบ** git pull คือรวมโค้ดจาก remote มายัง local โดยที่เราไม่สามารถรู้ได้เลยว่าจะรวมโค้ดอะไรบ้าง รู้แค่หลังจาก pull เสร็จแล้วนั่นเอง ซึ่งจริงๆแล้ว git pull มันก็คือการทำ git fetch และต่อด้วย git merge อัตโนมัติ

20. **ตอบ**