11.โปรแกรม version control มีประโยชน์อย่างไร

ช่วยให้คุณสามารถย้อนไฟล์บางไฟล์หรือแม้กระทั่งทั้งโปรเจคกลับไปเป็นเวอร์ชั่นเก่าได้ และยังจะช่วยให้คุณเปรียบเทียบการแก้ไขที่เกิดขึ้นในอดีต ดูว่าใครเป็นคนแก้ไขคนสุดท้ายที่อาจทำให้เกิดปัญหา แก้ไขเมื่อไร ฯลฯ และยังช่วยให้คุณสามารถกู้คืนไฟล์ที่คุณลบหรือทำเสียโดยไม่ตั้งใจได้อย่างง่ายดาย

12.ข้อได้เปรียบของ Distributed Version Control เมื่อเทียบกับ Centralized Version Control

ได้เปรียบกว่าตรงที่Centralized Version Control เป็นรวมศูนย์กลาง หากล่มก็จะเสียหายเหมือนกันเพราะทุกอย่างรวมกันอยู่ที่เซิร์ฟเวอร์ที่เดียว ถ้าเซิร์ฟเวอร์นั้นล่มชักชั่วโมงนึง หมายความว่าในชั่วโมงนั้นไม่มีใครสามารถทำงานร่วมกันหรือบันทึกการเปลี่ยนแปลงงานที่กำลังทำอยู่ไปที่เซิร์ฟเวอร์ได้เลย แต่Distributed Version Control เป็นแบบแบบกระจายศูนย์ แม้ว่าเซิร์ฟเวอร์จะเสีย client ก็ยังสามารถทำงานร่วมกันได้ต่อไปได้

13.ข้อได้เปรียบของ Centralized Version Control เมื่อเทียบกับ Distributed Version Control

มีเซิร์ฟเวอร์กลางที่เก็บไฟล์ทั้งหมดไว้ในที่เดียวและผู้ใช้หลาย ๆ คนสามารถต่อเข้ามาเพื่อดึงไฟล์จากศูนย์กลางนี้ไปแก้ไขได้ ซึ่งทุกคนสามารถรู้ได้ว่าคนอื่นในโปรเจคกำลังทำอะไร ผู้ควบคุมระบบสามารถควบคุมได้อย่างละเอียดว่าใครสามารถแก้ไขอะไรได้บ้าง การจัดการแบบรวมศูนย์ในที่เดียวทำได้ง่ายกว่าการจัดการฐานข้อมูลใน client แต่ละเครื่องเยอะ และ Distributed Version Control จะแชร์เฉพาะที่เปลี่ยนแปลงและไม่ด้ส่งกลับให้central ส่งให้เฉพาะคนที่อยากเห็น

14.โดยการแตก branch แรกเพื่อเพิ่มความสามารถให้โปรแกรมรับ -v สำหรับบอกเวอร์ชั่นโปรแกรม เช็คและแก้โค๊ด

15.พยายามอย่าใช้งานพร้อมกัน

16. Git คืออะไร แตกต่างจาก Githubอย่างไร

Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็คเรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของโปรเจ็คที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ ฉะนั้น Version Control ก็เหมาะอย่างยิ่งสำหรับนักพัฒนาไม่ว่าจะเป็นคนเดียวโดยเฉพาะอย่างยิ่งจะมีประสิทธิภาพมากหากเป็นการพัฒนาเป็นทีม ส่วนGitHub คือเว็บที่ให้บริการพื้นที่จัดเก็บโครงการโอเพ่นซอร์สด้วยระบบควบคุมเวอร์ชั่นแบบ Git โดยมีจุดประสงค์หลักคือ ทำให้การแบ่งปันและพัฒนาโครงการต่างๆด้วยกันเป็นไปได้ง่ายๆ

17.จุดประสงค์หลักในการ branch คืออะไร

เหมือนกับการก้อปปี้โค้ดที่อยู่ภายใน master ทั้งหมดไปเป็นอีกโฟลเดอร์หนึ่งแล้วตั้งชื่อใหม่ ทุกคนที่ร่วมกันทำงานก็ให้ตกลงกันว่าจะ commit, push โค้ดลงไปที่นี่ ทดสอบโค้ดกันที่นี่ พอหลังจากพัฒนาโค้ดจนดี ไม่มีบัก อยากจะขึ้นโปรดักชั่นแล้ว ค่อย merge ซึ่งbranch ไม่ใช่เพียงเพื่อแตกออกมาให้ master มันมีแต่โค้ดที่ปราศจากบักเท่านั้น แต่เรายังสามารถแตก branch ออกมาเพื่อทดลองเขียนโค้ดส่วนตัว ว่าถ้าเขียนอย่างนี้ได้ไหม เราก็แตก branch ออกมา หลังจากเล่นกับโค้ดจนพอใจแล้ว เราก็ลบ branch นั้นทิ้งไปก็ได้

18.การที่ Fast-Forward จะสลับ Head ไปทำให้มันไม่มีการ Conflict

19.คือรวมโค๊ดจาก remote มายัง local โดยที่เราไม่สามารถรู้ได้เลยว่าจะรวมโค๊ดอะไรบ้าง รู้แค่หลังจาก pull เสร็จแล้วนั่นเอง ซึ่งจริงๆแล้ว git pull มันก็คือการทำ git fetch และต่อด้วย git merge อัตโนมัตินั่นเอง

20.

เป็นversion control system ที่มีการพัฒนาเป็นversionต่างๆ โดยจะเก็บข้อมูลโดยเริ่มจาก List ของ Files ต้นฉบับ จากนั้นก็จะคอยเก็บบันทึกการเปลี่ยนแปลงที่เกิดขึ้นในแต่ละ Version