11. โปรแกรม version control มีประโยชน์ คือ

1. ช่วยในการลดการเก็บข้อมูล

2 .สามารถเรียกเวอร์ชั่นใดเวอร์ชั่นหนึ่งกลับมาดูเมื่อไรก็ได้

3. รู้ได้ว่าก่อนว่าก่อนหน้ามีการแก้ไขอะไรไปบ้าง

4.ข้อมูลไม่หายถ้ามีการบันทึกทับ

12. ข้อได้เปรียบของ distributed version control เมื่อเทียบกับ centralized version control คือสามารถทำ commit ได้แบบ offline เลยเพราะเป็นการ commit ไปที่ local repo. ในกรณี clone remote repo. มาแล้ว ก็เท่ากับว่าเรามี local repo. ที่เหมือนกันกับ remote repo.

13. ข้อได้เปรียบของ centralized version control เมื่อเทียบกับ distributed version control คือทุกอย่างจะถูกเก็บในตัวกลาง ซึ่งก็คือ remote repo. ทั้งหมด จึงง่ายต่อการเรียกใช้

14. แนวทางในการแก้ไข conflict ที่เกิดขึ้นเมื่อมีการ merge โปรแกรมของผู้พัฒนาหลายๆ คนเข้าด้วยกัน คือ ให้ทำการ merge บ่อย ๆ ไปเลยนั่นคือ ทุกครั้งเมื่อคุณทำการเปลี่ยนแปลง หรือ commit source code นั่นเองจะช่วยลดข้อขัดแย้งต่าง ๆ ลงไปอย่างมากถึงจะเกิดข้อขัดแย้ง ก็เป็นเพียงปัญหาเล็ก ๆซึ่งสามารถแก้ไขได้อย่างง่ายดาย

15. แนวทางในการลด conflict ที่เกิดขึ้นเมื่อมีการ merge โปรแกรมของผู้พัฒนาหลายๆ คนเข้าด้วยกัน คือ ลดการชนหรือขัดแย้งกันมากเหลือเกินของจำนวน source code

16. Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็คเรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของโปรเจ็คที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้

แตกต่างจาก Github คือ เว็บที่ให้บริการพื้นที่จัดเก็บโครงการโอเพ่นซอร์สด้วยระบบควบคุมเวอร์ชันแบบ Git ดังนั้น Git ทำงานบน Github นั่นเอง

17. จุดประสงค์หลักในการ branch คือ ช่วยให้นักพัฒนาสามารถที่จะทำงานได้สะดวกขึ้น(ช่วยให้เรารู้ว่าเราอยู่ที่ตำแหน่งไหน) ยกตัวอย่างเช่น เรามีโค๊ดที่ดีอยู่แล้ว แต่อยากจะทดลองอะไรนิดๆหน่อย หรือแก้ไขอะไรก็ตาม ไม่ให้กระทบกับตัวงานหลัก ก็เพียงแค่สร้าง branch ใหม่ขึ้นมา เมื่อแก้ไขหรือทำอะไรเสร็จแล้ว ก็ค่อยเซฟกลับมาที่ master เหมือนเดิม

18. fast forward merge คือ การ merge commit เข้ากับ commit ที่สามารถไปถึงได้โดยการตาม history ของ commit อันแรก Git จะทำให้ทุกอย่างง่ายขึ้นโดยการขยับ pointer ไปข้างหน้าเพราะมันไม่มีงานที่ถูกแยกออกไปให้ merge

การ push ไปที่ remote repo ควรใช้ fast forward merge เพราะ Git จะทำให้ทุกอย่างง่ายขึ้นโดยการขยับ pointer ไปข้างหน้า

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

20.วิธีที่ Git มองข้อมูลต่าง ๆ โดยทั่วไประบบอื่นมักจะเก็บข้อมูลในรูปแบบของการแก้ไขที่เกิดขึ้นกับไฟล์ต่าง ๆ ระบบเหล่านี้ (เช่น CVS, Subversion, Perforce, Bazaar, ฯลฯ) จะมองข้อมูลในรูปแบบของไฟล์และการแก้ไขต่าง ๆ ที่เกิดขึ้นกับไฟล์แต่ละไฟล์