11. - มีประโยชน์เหนือกว่า local VCS ในหลายด้าน เช่น สามารถดูได้ว่าคนอื่นในโปรเจคกำลังทำอะไร ผู้ควบคุมสามารถควบคุมได้ละเอียดว่าใครแก้อะไรได้บ้าง

12. - distributed version control จึงเร็วกว่า และผู้ใช้งานก็สามารถแก้ได้ทุกที่ โดยไม่จำเป็นจะต้องติดต่อกับ central repo ก็ได้

13. - centralized version control ทุกอย่างจะเก็บไว้ในตัวกลาง จำเป็นต้องเชื่อมต่อ Network

14. - ข้อมูลชนกันทำให้เกิน error ส่งไม่สำเร็จ

15. - ไม่ควร merge พร้อมๆกัน

- แต่ละงานที่ทำควรเป็นงานชิ้นเล็กๆ

- แต่ละงานที่ทำ Code ควรแยกออกจากกันชัดเจน

16. - **Git** คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็คเรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของโปรเจ็คที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ ส่วน Github เป็นเว็บเซิฟเวอร์ที่ให้บริการในการฝากไฟล์ Git

17. - คือการแบ่งหรือแตกกิ่ง จากใหญ่แล้วแบ่งเป็นย่อยๆ เป็น feature ที่ช่วยให้นักพัฒนาสามารถที่จะทำงานได้สะดวกขึ้น เช่น เรามี Code ดีแล้วแต่อยากแก้ไขเพิ่ม เพื่อไม่ให้กระทบกับงานหลัก แค่สร้าง branch ใหม่ขึ้นมา แล้วค่อย Save กลับมา master เดิม

18. -  เพราะ commit ที่ถูกชี้โดย branch ที่เรา merge เป็น upstream ของ commit ที่เราอยู่โดยตรง Git ก็เลยขยับ pointer ไปข้างหน้า อีกนัยหนึ่งก็คือ เวลาที้ราพยายามจะ merge commit ซักอันเข้ากับ commit ที่สามารถไปถึงได้โดยการตาม history ของ commit อันแรก Git จะทำให้ทุกอย่างง่ายขึ้นโดยการขยับ pointer ไปข้างหน้าเพราะมันไม่มีงานที่ถูกแยกออกไปให้ merge

19. - จริงๆแล้ว git pull ก็คือรวมโค๊ดจาก remote มายัง local โดยที่เราไม่สามารถรู้ได้เลยว่าจะรวมโค๊ดอะไรบ้าง รู้แค่หลังจาก pull เสร็จแล้วนั่นเอง ซึ่งจริงๆแล้ว git pull มันก็คือการทำgit fetch และต่อด้วย git merge อัตโนมัติ