ข้อ 11. ช่วยให้สามารถย้อนไฟล์บางไฟล์หรือแม้กระทั่งทั้งโปรเจคกลับไปเป็นเวอร์ชั่นเก่าได้ นอกจากนั้นระบบ VCS ยังจะช่วยเปรียบเทียบการแก้ไขที่เกิดขึ้นในอดีต ดูว่าใครเป็นคนแก้ไขคนสุดท้ายที่อาจทำให้เกิดปัญหา แก้ไขเมื่อไร ฯลฯ และยังช่วยให้สามารถกู้คืนไฟล์ที่ลบหรือทำเสียโดยไม่ตั้งใจได้อย่างง่ายดาย

ข้อ 12. distributed version control เร็วกว่า และผู้ใช้งานก็สามารถแก้ไขได้ทุกที่ โดยไม่จำเป็นจะต้องติดต่อกับ central repo ก็ได้

ข้อ 13. เปรียบเทียบความแตกต่างของไฟล์เวอร์ชันต่างๆ ได้ เราสามารถกลับไปดูได้ว่ามีอะไรเปลี่ยนแปลงไปบ้างในแต่ละเวอร์ชันของไฟล์

ข้อ 14. ทำการ edit แล้ว commit ไปใหม่นั่นเอง

ข้อ 15. ปรึกษากันในทีม แล้วแบ่งไฟล์ที่จะนำไปแก้ไขให้ระบุ บุคคลให้ชัดเจน

ข้อ 16. Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็คเรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของโปรเจ็คที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข แตกต่างจาก GitHub คือ เว็บที่ให้บริการพื้นที่จัดเก็บโครงการโอเพ่นซอร์สด้วยระบบควบคุมเวอร์ชันแบบ Git โดยมีจุดประสงค์หลักคือ ทำให้การแบ่งปันและพัฒนาโครงการต่างๆด้วยกันเป็นไปได้ง่ายๆ

ข้อ 17. อยากจะทดลองแก้ Bug หรือแก้ไขอะไรก็ตาม ไม่ให้กระทบกับตัวงานหลัก ก็เพียงแค่สร้าง branch ใหม่ขึ้นมา พอถีงเวลาที่เราเสร็จการแก้ master branch แล้ว เราสามารถใช้คำสั่ง merge เพื่อรวมงานของ masterได้

ข้อ 18. ในการ Merge Branch บน Git นั้น หาก Commit สุดท้ายของ Branch ปลายทาง เป็น Commit เดียวกับจุดที่แยก Branch ออกมา ส่วนการ push ไปที่ remote repo จึงควรเป็น forward merge ก็คือ เพราะต้อง merge ไปรวมกับ master นั่นเอง

ข้อ 19. คือรวม Code จาก remote มายัง local โดยที่เราไม่สามารถรู้ได้เลยว่าจะรวมโค๊ดอะไรบ้าง รู้แค่หลังจาก pull เสร็จแล้วนั่นเอง ซึ่งจริงๆแล้ว git pull มันก็คือการทำ git fetch และต่อด้วย git merge อัตโนมัตินั่นเอง

.ข้อ 20. version control git