11. ระบบที่จัดเก็บการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์หนึ่งหรือหลายไฟล์เพื่อที่คุณสามารถเรียกเวอร์ชั่นใดเวอร์ชั่นหนึ่งกลับมาดูเมื่อไรก็ได้ หนังสือเล่มนี้จะยกตัวอย่างจากไฟล์ที่เป็นซอร์สโค้ดของซอฟต์แวร์ และสามารถใช้ version control กับไฟล์ชนิดใดก็ได้

12. Distributed Version Control Systems (DVCSs) แม้ว่าเซิร์ฟเวอร์จะเสีย client ก็ยังสามารถทำงานร่วมกันได้ต่อไป และ repository เหล่านี้ของ client ยังสามารถถูกก๊อปปี้กลับไปที่เซิร์ฟเวอร์เพื่อกูข้อมูลกลับคืนก็ได้ แต่ Centralized Version Control Systems (CVCSs) ถ้าข้อมูลเสียข้อมูลจะมีความเสียหายเพราะรวมข้อมูลไว้แค่ส่วนกลาง

13. Centralized Version Control Systems (CVCSs) มีเซิร์ฟเวอร์กลางที่เก็บไฟล์ทั้งหมดไว้ในที่เดียวและผู้ใช้หลาย ๆ คนสามารถต่อเข้ามาเพื่อดึงไฟล์จากศูนย์กลางนี้ไปแก้ไขได้ แต่ Distributed Version Control Systems (DVCSs) จะไม่สามารถดึงข้อมูลได้และแก้ไขได้นั่นเอง

14.เช่น ต้องไม่ตำหนิว่า ทำไมคุณจึงพยายามที่จะสร้างปัญหาอยู่ตลอดเวลากับผม

15.มีวิธีลดอยู่

            1.   การหลีกเลี่ยง (Avoidance)

            2.   การปรองดอง (Accommodation)

            3.   การประนีประนอม (Compromise)

            4.   การแข่งขัน (Competition)

            5.   การร่วมมือกัน (Collaboration)

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

**Github** คืออะไร ? Github เป็นเว็บเซิฟเวอร์ที่ให้บริการในการฝากไฟล์ Git (ทั่วโลกมักนิยมใช้ในการเก็บโปรเจ็ค Open Source ต่างๆ ที่ดังๆ ไม่ว่าจะเป็น Bootstrap, Rails, Node.js, Angular เป็นต้น)

ความแตกต่างนั้นก้คือ Git สามารถเก็บงานของเราไว้ดูได้คนเดียว ส่วน Github นั้นเป็นการเก็บงานแต่สามารถดูได้หลายคนซึ่งเป็นเอาการขึ้นเซิฟเวิร์ จึงเป็นที่นิยมต่อคนในปัจจุบัน

17. จุดประสงค์ เป็น feature ที่ช่วยให้นักพัฒนาสามารถที่จะทำงานได้สะดวกขึ้น ยกตัวอย่างเช่น เรามีโค๊ดที่ดีอยู่แล้ว แต่อยากจะทดลองอะไรนิดๆหน่อย หรือแก้ไขอะไรก็ตาม ไม่ให้กระทบกับตัวงานหลัก ก็เพียงแค่สร้าง branch ใหม่ขึ้นมา เมื่อแก้ไขหรือทำอะไรเสร็จแล้ว ก็ค่อยเซฟกลับมาที่ master เหมือนเดิม

18.เป็นการ commit messageที่เคย commit ไว้ก็จะถูกรวมมาอยู่ใน master หมด ทีนี้เมื่อเราทำ feature อะไรเสร็จแล้ว brach

19. git pull ก็คือรวมโค๊ดจาก remote มายัง local โดยที่เราไม่สามารถรู้ได้เลยว่าจะรวมโค๊ดอะไรบ้าง รู้แค่หลังจาก pull เสร็จแล้วนั่นเอง ซึ่งจริงๆแล้ว git pull มันก็คือการทำ git fetch และต่อด้วยgit merge อัตโนมัตินั่นเอง

20.สื่อให้เห็นว่าการทำงานมีลดับเป็นขั้นเป็นต่อและแต่ละชั้นนั้น การทำงานไม่เหมือนกัน