11.ตอบ

1. เวลา Commit งานเสร็จแล้วมันสบายใจแค่ไหน (ไม่คิดเรื่อง Bug ฮ่าๆ) เวลาที่เกิดอะไรผิดพลาดขึ้นมาแล้วสามารถ Revert กลับได้

2. มีการเก็บ code เก่าๆไว้ให้ด้วย ไม่ต้อง สำเนาเองให้วุ่นวายใจ

3. มีการบอกด้วยว่า ลบอะไรเข้าไป เพิ่มอะไรเข้ามา

4.code จะไม่หาย แล้วอัพโหลดเอาไว้บน github ก็ไม่ต้องกลัวเครื่องเจ๊ง เพราะมันอยู่บน internet จะได้มีงานเอาไว้ดู

12. ตอบ สามารถย้อนไฟล์บางไฟล์หรือแม้กระทั่งทั้งโปรเจคกลับไปเป็นเวอร์ชั่นเก่าได้ นอกจากนั้นระบบ VCS ยังจะช่วยให้คุณเปรียบเทียบการแก้ไขที่เกิดขึ้นในอดีต ดูว่าใครเป็นคนแก้ไขคนสุดท้ายที่อาจทำให้เกิดปัญหา แก้ไขเมื่อไร ฯลฯ และยังช่วยให้คุณสามารถกู้คืนไฟล์ที่คุณลบหรือทำเสียโดยไม่ตั้งใจได้อย่างง่ายดาย

13. ตอบ การกระทำของโคลนพื้นที่เก็บข้อมูลทั้งหมดให้กระจายเครื่องมือในการควบคุมรุ่นข้อดีกว่าระบบส่วนกลาง:

- การกระทำการแสดงอื่น ๆ นอกเหนือจากการผลักดันและดึงการแก้ไขเป็นไปอย่างรวดเร็วมากเพราะเครื่องมือเฉพาะความต้องการในการเข้าถึงฮาร์ดไดรฟ์ไม่ได้เป็นเซิร์ฟเวอร์ระยะไกล

- การกระทำการแก้ไขใหม่สามารถทำได้ทั้งในประเทศโดยไม่ต้องให้คนอื่นได้เห็นพวกเขา เมื่อคุณมีกลุ่มของการแก้ไขพร้อมคุณสามารถผลักดันให้พวกเขาทั้งหมดในครั้งเดียว

- ทุกอย่าง แต่การผลักดันและดึงสามารถทำได้โดยไม่ต้องเชื่อมต่ออินเทอร์เน็ต เพื่อให้คุณสามารถทำงานบนเครื่องบินและคุณจะไม่ถูกบังคับให้ฆ่าหลาย bugfixes เป็นเซ็ตขนาดใหญ่

- เนื่องจากแต่ละโปรแกรมเมอร์มีฉบับเต็มของพื้นที่เก็บข้อมูลโครงการที่พวกเขาสามารถแบ่งปันการเปลี่ยนแปลงที่มีหนึ่งหรือสองคนอื่น ๆ ได้ตลอดเวลาหากพวกเขาต้องการที่จะได้รับข้อเสนอแนะบางก่อนที่จะแสดงการเปลี่ยนแปลงให้กับทุกคน

14.ตอบ

15. ตอบ

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

ถ้าไม่มี Github การจะทำงานร่วมกันกับ Git จะต้องพิมพ์ผ่าน command หรือ Terminal เท่านั้น ดังนั้น Github จึงถูกพัฒนาขึ้นมาเพื่อทำให้การทำงานกับ Git สะดวกและง่ายดายมากขึ้น Github คือ application ทำงานร่วมกันกับ Git โดยมี interface ที่ทำใช้งานง่าย

17. ตอบ การก้อปปี้โค้ดที่อยู่ภายใน master ทั้งหมดไปเป็นอีกโฟลเดอร์หนึ่งแล้วตั้งชื่อใหม่

18. ตอบ commit ที่ถูกชี้โดย branch ที่คุณ merge มันเป็น upstream ของ commit ที่คุณอยู่โดยตรง Git ก็เลยขยับ pointer ไปข้างหน้า พูดอีกนัยหนึ่งก็คือ เวลาที่คุณพยายามจะ merge commit ซักอันเข้ากับ commit ที่สามารถไปถึงได้โดยการตาม history ของ commit อันแรก Git จะทำให้ทุกอย่างง่ายขึ้นโดยการขยับ pointer ไปข้างหน้าเพราะมันไม่มีงานที่ถูกแยกออกไปให้ merge

19. ตอบ ใช้ดึงความเปลี่ยนแปลงจาก remote มายัง local และรวมเข้าด้วยกัน (มีค่าเท่ากับ fetch+merge)

20.