

11.คอยจัดการ backup source code ของเรา โดยเก็บเป็นลักษณะ version ต่างๆ เช่นในกรณีเกิดปัญหาขึ้น ก็สามารถหยิบเอา source code ตัวเก่าที่เคยใช้งานได้มาแทน ซึ่งคล้ายๆ กับการทำ Undo งาน แต่ว่า Version Control ยังเป็นตัวกลางที่ทำให้ source code ของแต่ละเครื่อง (แต่ละ programmer) มี source code ที่ตรงกัน ซึ่งโดยปกติทั่วไปแล้ว programmer มักจะทำโปรแกรมให้เสร็จสมบูรณ์ และใช้งานได้ก่อน จึงค่อยโยนขึ้นไป repository เพื่อให้คนอื่นดึงไปใช้ต่อไป

12. distributed version control มีข้อได้เปรียบกว่าตรงที่ แต่ละคนจะมี copy ไฟล์ตั้งแต่เริ่มต้นจนท้ายสุด เพราะแบบนั้นแต่ละคนก็สามารถ maintain code ได้โดยที่ทุกคนมี copy ของไฟล์ทั้งหมดอยู่ที่ local computer และ change set ของแต่ละ version ที่เกิดจากหลายๆ ทีมนั้นสามารถรวมกันได้ ไม่จำเป็นต้องรอ change copy จาก central repo เหตุนี้จึงทำให้ distributed version control จะเร็วกว่า และสามารถแก้ไขได้ทุกที่โดยไม่จำเป็นต้องติดต่อ central repo ก็ได้

13. centralized version control มีข้อได้เปรียบตรงที่ change ของไฟล์ทั้งหมดจะถูกเก็บไว้อยู่ central repo ที่เดียวดังนั้นจึงทำให้ทุกคนในทีมสามารถเชื่อมต่อตรงกันได้ทั้งหมด

14.อย่างแรกควรทำการ merge บ่อยๆ เพราะ ทุกครั้งที่ทำการ commit source code จะช่วยลดข้อขัดแย้งต่างๆ ลงไปอย่างมาก อย่างที่สองคือ การออกแบบที่ดีเพราะช่วยให้ทีมสามารถทำงานร่วมกันได้ง่าย อย่างที่สามคือ การพูดคุยการสื่อสาร เป็นสิ่งสำคัญเพื่อบอกถึงการทำงานที่กำลังทำอยู่ในขณะนั้น อย่างสุดท้ายคือ การทำงานด้วยกันจะช่วยแก้ปัญหา Merge conflict ได้ 100 เปอร์เซ็นต์ เนื่องจากทุกคนมานั่งทำงานร่วมกัน หรือใช้คำเครื่องเดียวกัน จะทำให้ไม่มีทางที่ source code จะขัดแย้งกัน (แต่มีค่าใช้จ่ายสูง)

15.การทำงานในแต่ละงานควรจะเป็นงานที่ code แยกออกจากกันอย่างชัดเจน ไม่เช่นนั้นหากทำ code เดียวกันจะทำให้เกิดโอกาส merge conflict กันสูง และการช่วยลดที่ดีอีกทางหนึ่งคือ งานแต่ละงานที่ทำนั้นควรเป็นงานเล็กๆ เพราะยิ่งใหญ่มากจะทำให้เกิด local branch ที่ยาวนานเกินไป

16. **Git** คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ ในโปรเจกต์เรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชันต่างๆของโปรเจกต์ที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ ฉะนั้น Version Control ก็เหมาะสมอย่างยิ่งสำหรับนักพัฒนา ไม่ว่าจะเป็นคนเดียวโดยเฉพาะอย่างยิ่งจะมีประสิทธิภาพมากหากเป็นการพัฒนาเป็นทีม ส่วน GitHub คือเว็บไซต์ที่ให้บริการเสมือน git server แต่ github นั้นจะเป็นการปล่อยแบบ public ซึ่งคนอื่นสามารถมาดู source code ของเราได้ และถ้าหากต้องการ แบบ private ก็ต้องเสียตังในการซื้อ

17.เป็นการแยกการทำงานออกมาเพื่อไม่ให้กระทบกับการทำงานหลัก เมื่อแก้ไขหรือ ทำเสร็จแล้ว ก็ save แล้วกลับมาที่ master เหมือนเดิม

18.คือ หมายเลข commit ของ branch master ถูกนำไปใช้ตัวเดียวกับ merge

19.คือการรวม code จาก remote มายัง local