57160176 นายธนพล สิงห์ทอง กลุ่ม 1

11. ช่วยจัดเก็บข้อมูลการเปลี่ยนแปลงของไฟล์แต่ละ version และสามารถเรียกไฟล์ที่เก็บไว้ของแต่ละ version ออกมาแสดงได้เสมอ

12. แบบ distributed version control สามารถกระจายไฟล์ไว้หลายๆที่ทำให้เวลาที่ serverเสียหรือไม่สามารถใช้งานได้นั้น client ก็ยังดำเนินงานได้ต่อซึ่งต่างจากแบบ centralized version control ถ้า server ล่ม client ก็ไม่สามารถเข้าถึงไฟล์ได้เลย

13. centralized version control ดีกว่าตรงที่ง่ายในการจัดการฐานข้อมูลมากกว่าและ ผู้คุมระบบนั้นสามารถกำหนดได้ว่าใครมีสิทธิแก้ไขข้อมูลส่วนใดได้บ้าง และรู้ว่าแต่ละ client ทำอะไรอยู่บ้าง

14. การแก้ไขโดยตรงคือาการใช้โปรแกรมเข้าช่วย merge source code ซึ่งเป็นวิธีแก้ไขไม่ใช่การป้องกัน , Mob programming คือมารุมทำงานด้วยกันกันบนเครื่องเดียวกัน

15. การลดปัญหาการ merge conflict คือต้อง merge กันถี่ขึ้นเพราะถ้าทิ้งไว้นาน conflict ก็จะยิ่งบานปลายขึ้น , ใช้ 1 class 1 method ก็จะดีทำให้เราตรวจสอบได้ง่าย , ต้องมีการติดต่อกันภายในทีม

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

Github คือเว็บไซต์บริการจัดเก็บข้อมูลด้วยระบบควบคุม version แบบ Git

17. เพื่อคัดลอกโค้ดที่อยู่ในไฟล์หลักไปอีกโฟล์เดอร์หนึ่งอล้วตั้งชื่อใหม่ แล้วสมาชิกทีมจะแก้ไขในส่วนนี้ ทดสอบกันที่นี่ พัฒนาดันจนสมบูรณ์แล้วก็จะ merge กลับไปยังไฟล์หลัก

18. merge แบบ fast forward จะเป็นแบบ merge รวม branch ไปเป็นสายเดียวกันเลยจะทำให้เราดูไม่ออกว่า branch มาจากแยกออกมาจุดไหน การที่ push ไปที่ remote repo ใช้ fast forward merge นั้นเพราะ ไม่ได้ branch มาจากที่ได้และไม่ต้องกันรู้จุดเริ่มและรวมกลับของการ branch

19. สำหรับดึงการเปลี่ยนแปลงล่าสุดจาก remote repo มาที่ local repo

20. การแตก branch และการ merge ของโปรเจคหนึ่ง