11. โปรแกรม version control มีประโยชน์อย่างไร

มันทำแบ้กอัพให้เราเอง ถ้าเราทำอะไรผิดขึ้นมา เราสามารถจะย้อนมันกลับไปตอนที่ยังทำงานได้ดีๆ อย่างสบายๆ

12.ข้อได้เปรียบของ distributed version control เมื่อเทียบกับ centralized version control คืออะไร

ทั้ง SCCS, RCS, CVS, SVN นั้นใช้ central code repository model คือหมายความง่ายๆ ว่า changes ของ file ทั้งหมดถูกเก็บอยู่ที่ central repo หรือศูนย์กลางที่เดียว ถ้าสมมติว่า central repo

เกิดพังขึ้นมาหรือผู้ใช้ไม่สามารถติดต่อกับ central repo ได้ก็จะทำให้ไม่สามารถดึงไฟล์ที่พึ่งเปลี่ยนแปลงมาทีตัวเองได้ ซึ่ง centralized version control นั้นเก็บข้อมูลแค่ที่ที่เดียว

แต่ distributed version control นั้นใช่หลายตัวในการเก็บ

13.ข้อได้เปรียบของ centralized version control เมื่อเทียบกับ distributed version control คืออะไร

คือระบบ Version Control ที่ไม่มี Server หลักเป็นศูนย์กลาง แต่ละเครื่องของแต่ละคนก็จะเก็บข้อมูลของโปรเจคทั้งหมดไว้ อาจจะทำให้ยุ่งยากกว่าการที่มี server ใหญ่ server เดียว

14.บอกแนวทางในการแก้ไข conflict ที่เกิดขึ้นจากการ merge โปรแกรมของผู้พัฒนาหลายๆคนเข้าด้วยกัน

Feature Toggle , Revert Code

15.บอกแนวทางในการลด conflict ที่เกิดขึ้นจากการ merge โปรแกรมของผู้พัฒนาหลายๆคนเข้าด้วยกัน

1) ทำการ Merge บ่อย ๆ ทุกครั้งเมื่อคุณทำการเปลี่ยนแปลง หรือ commit source code นั่นเองจะ ช่วยลดข้อขัดแย้งต่าง ๆ ลงไปอย่างมาก

2) การพูดคุย การสื่อสาร มันสำคัญอย่างมากถ้าคุณไม่รู้ ก็ควรพูดคุยกันเพื่อให้รู้ถึงโปรแกรมของกันและกัน หรือบางครั้งต้องแก้ไข class เดียวกันอยู่ตลอดเวลา

3) Mob programming ทุกคนมานั่งทำงานด้วยกันใช้เครื่องทำงานเดียวกัน ดังนั้น ไม่มีทางที่ source code จะขัดแย้งกัน

16.git คืออะไร แตกต่างจาก github อย่างไร

Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็คเรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของโปรเจ็คที่ใด

เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ ฉะนั้น Version Control ก็เหมาะอย่างยิ่งสำหรับนักพัฒนาไม่ว่าจะเป็นคนเดียว

โดยเฉพาะอย่างยิ่งจะมีประสิทธิภาพมากหากเป็นการพัฒนาเป็นทีม GitHub คือ เว็บที่ให้บริการพื้นที่จัดเก็บโครงการโอเพ่นซอร์สด้วยระบบควบคุมเวอร์ชันแบบ Git โดยมีจุดประสงค์หลักคือ

ทำให้การแบ่งปันและพัฒนาโครงการต่างๆด้วยกันเป็นไปได้ง่ายๆ

17.จุดประสงค์หลักในการ branch คืออะไร

เวลาคุณ commit ใน Git นั้น Git จะเก็บเป็น object ของการ commit ซึ่งประกอบด้วย pointer ชี้ไปยัง snapshot ของ content ที่คุณ stage ไว้, metadata ของชื่อผู้แก้ไขและ message ที่บันทึกไว้ และ pointer ที่ชี้ไปยัง parent

ลำดับถัดไปของ commit นั้นๆ (ซึ่งอาจจะไม่มีก็ได้ถ้าเป็น commit ครั้งแรก, อาจจะมีอันเดียวชี้ไปยัง parent ของ commit ปรกติทั่วไปหรืออาจจะมี parent หลายอันสำหรับ commit ที่เป็นผลจากการ merge หลายๆ branch เข้าด้วยกัน)

18.Fast forward merge คืออะไรและทำไมการ push ไปที่ remote repo จึงควรต้อง merge แบบนี้

Fast Forward Effect คือการที่แตก Branch ออกไปทำงาน และใช้การ Merge เข้ามาที่ Branch นั้น และ Git คำนวณแล้วว่า

ไม่มีการเกิด Conflict ระหว่างไฟล์นั้นก็ตามในการ Merge จึงสามารถนำ commit ที่เกิดขึ้นใน Branch ที่แตกออกไปทำงาน มาต่อได้เลย

19.หน้าที่หลักของคำสั่ง git pull คืออะไร

การรวมcodeจาก remote มายัง local โดยที่เราไม่สามารถรู้ได้เลยว่าจะรวมcodeอะไรบ้าง จะรู้แค่หลังจาก pull เสร็จแล้ว

20.แผนภาพด้านล่างนี้ต้องการสื่อความหมายอะไร

แผนภาพในการ Coding โดยแสดงให้เห็นว่าช่วงไหนทำอะไรบ้าง Release , Development , Feature มีการ Merge หรือ Branch ตอนไหน