พื้นฐานเกี่ยวกับ version control และ git

**11.โปรแกรม version control มีประโยชน์อย่างไร**

ระบบที่คอยจัดการ backup source code ของเรา โดยเก็บเป็นลักษณะ version ต่างๆ เช่นในกรณีเกิดปัญหาขึ้น ก็สามารถหยิบเอา source code ตัวเก่าที่เคยใช้งานได้มาแทน Version Control ตัวที่เด่นดังอยู่ตอนนี้ก็มีอยู่ 2 ตัว คือ Subversion(SVN) และ Git

**12.ข้อได้เปรียบของ distributed version control เมื่อเทียบกับ centralized version control คืออะไร**

ทำงานกับหลาย ๆ repository ได้อย่างดี ทำให้คุณสามารถทำงานกับคนหลายกลุ่มซึ่งทำงานในรูปแบบต่างกันในโปรเจคเดียวกันได้อย่างง่ายดาย เนื่องจากระบบเหล่านี้สนับสนุนการทำงานได้หลากหลายรูปแบบ ซึ่งอาจทำได้ยากในระบบแบบรวมศูนย์

**13.ข้อได้เปรียบของ centralized version control เมื่อเทียบกับ distributed version control คืออะไร**

การทำงานแบบ **centralized version control** ได้เปรียบ **distributed version control** ในหลายด้าน เช่น ทุกคนสามารถรู้ได้ว่าคนอื่นในโปรเจคกำลังทำอะไร ผู้ควบคุมระบบสามารถควบคุมได้อย่างละเอียดว่าใครสามารถแก้ไขอะไรได้บ้าง การจัดการแบบรวมศูนย์ในที่เดียวทำได้ง่ายกว่าการจัดการฐานข้อมูลใน client แต่ละเครื่องเยอะ

**14.บอกแนวทางการแก้ไข conflict ที่เกิดขึ้นเมื่อมีการ merge โปรแกรมของผู้พัฒนาหลายๆคนเข้าด้วยกัน**

**15.บอกเเนวทางในจการลด confict ที่เกิดขึ้นจากกา merge โปกรเเกรมของผู้พัฒนาหลายๆคนเข้าด้วยกัน**

เวลาที่ต้องการจะ merge commit ซักอันเข้ากับ commit ที่สามารถไปถึงได้โดยการตาม history ของ commit อันแรก Git จะทำให้ทุกอย่างง่ายขึ้นโดยการขยับ pointer ไปข้างหน้าเพราะมันไม่มีงานที่ถูกแยกออกไปให้ merge

**16.Git คืออะไร แตกต่างจาก Github อย่างไร**

เเตกต่างเพราะ Git เป็นระบบ Version Control เป็นระบบที่จัดการกับการเปลี่ยนแปลงของไฟล์ที่เกิดขึ้น ซึ่งจะมีการเก็บข้อมูลเป็นเวอร์ชัน สามารถเรียกดูประวัติได้ หรือสามารถย้อนกลับไป Source Code ก่อนหน้าได้ ซึ่ง Git จะใช้การเก็บข้อมูลแบบกระจายศูนย์(Git จะเก็บไฟล์หรือข้อมูลไว้ที่ Local และเก็บไว้ที่ Server) และสามารถติดต่อกับ Server ได้ทั้งแบบ Online และ Offlin เเต่ Github นั้นเป็นเว็บที่ให้บริการพื้นที่จัดเก็บโครงการโอเพ่นซอร์สด้วยระบบควบคุมเวอร์ชันแบบ Git โดยมีจุดประสงค์หลักคือ ทำให้การแบ่งปันและพัฒนาโครงการต่างๆด้วยกันเป็นไปได้ง่ายๆ

**17.จุดประสงค์หลักในการ branch คืออะไร**

เพื่อให้นักพัฒนาสามารถที่จะทำงานได้สะดวกขึ้น ยกตัวอย่างเช่น เรามีโค๊ดที่ดีอยู่แล้ว แต่อยากจะทดลองอะไรนิดๆหน่อย หรือแก้ไขอะไรก็ตาม ไม่ให้กระทบกับตัวงานหลัก ก็เพียงแค่สร้าง branch ใหม่ขึ้นมา เมื่อแก้ไขหรือทำอะไรเสร็จแล้ว ก็ค่อยเซฟกลับมาที่ master เหมือนเดิม

**19.หน้าที่หลักของคำสั่ง git pull คืออะไร**

คือการรวมโค๊ดจาก remote มายัง local ซึ่งผูใข้จะไม่รู้เลยว่ารวมโค๊ดอะไรไว้บ้าง จะรู้เเค่หลังจาก pull เสร้จเเล้ว

20.แผนภาพด้านล่างนี้ต้องการสื่อความหมายใด



จากภาพ เราจะเห็นวิธีการตั้งเวอร์ชั่นของโปรแกรมที่เราเขียนด้วย ถ้าหากเป็นการเอาโค้ดจาก Release ขึ้นโปรดักชั่น เขาจะใช้หมายเลขเวอร์ชั่นใหญ่ เช่น 1.0, 2.0, 3.0 แต่ถ้าเป็นการ Hotfix เขาจะใช้เลขเวอร์ชั่นย่อย เช่น 1.1, 1.2, 2.1, 3.1 เป็นต้น