ธนภาค ประจุมาศ

57160178 กลุ่ม 2

11. โปรแกรม version control มีประโยชน์อะไรบ้าง

1. สามารถหยิบเอา source code ตัวเก่าที่เคยใช้งานได้มาแทน

2. เป็นตัวกลางที่ทำให้ source code ของแต่ละเครื่อง (แต่ละ programmer) มี source code ที่ตรงกัน

3. เวลาที่เกิดอะไรผิดพลาดขึ้นมาแล้วสามารถ Revert กลับได้

4. สามารถทำให้เราพัฒนาโปรแกรมได้อย่างมั่นใจ

5. ไม่เสียเงินเพราะเป็น Open Source

6. มีคนใช้เยอะ ถ้ามี Bug หรือ Error เราสามารถหาอ่านได้ง่าย

12. ข้อได้เปรียบของ distributed version control เมื่อเทียบกับ centralized version control คืออะไร

- กระจายความเสี่ยงของ Project History จะหายได้ เพราะพังไปเครื่องหนึ่งเครื่องอื่นๆ ก็ยังมีอยู่ไม่หายไปไหนต่างกับ Centralized Version Control ถ้า Server เสีย Project History ทั้งหมดก็จะหายไปหมด

13. ข้อได้เปรียบของ centralized version control เมื่อเทียบกับ distributed version control คืออะไร

- มี Server หลักเป็นศูนย์กลางในการเก็บข้อมูลของโปรเจคทั้งหมด Client แต่ละเครื่องจะเป็นแค่ Working Copy เท่านั้นต่างกับ distributed version control ที่ไม่มี Server หลักเป็นศูนย์กลาง

14. บอกแนวทางในการแก้ไข conflict ที่เกิดขึ้นเมื่อมีการ merge โปรแกรมของผู้พัฒนาหลายๆคนเข้าด้วยกัน

- การพูดคุย การสื่อสารกัน ควรให้แต่ละคนในทีมได้พูดคุยกันว่าแต่ละคนทำงานอะไร รู้หรือไม่ว่าแต่ละคนแก้ไข class อะไรกันอยู่ รู้หรือไม่ว่า สิ่งที่แก้ไขไปนั้นกระทบใครบ้าง ดังนั้น ถ้าไม่รู้ก็ควรพูดคุยกันให้เข้าใจ เราสามารถแก้ไขการเกิด conflict ได้ง่ายๆเพียงแค่เราคุยกันในเรื่องที่ควรจะคุยกันมากขึ้น

15. บอกแนวทางในการลด conflict ที่เกิดขึ้นเมื่อมีการ merge โปรแกรมของผู้พัฒนาหลายๆคนเข้าด้วยกัน

- **Single Responsibility Principle (SRP)** เป็นหนึ่งในแนวทางการออกแบบระบบงานที่ดี คือ [SOLID](http://www.somkiat.cc/testable-design-part-2-about-solid/)แบ่งเป็นส่วนเล็กๆ คือ ในหนึ่ง class ในหนึ่ง method นั้นควรจะมีหน้าที่การทำงานเพียงอย่างเดียวเท่านั้น หรือในแต่ละ class แต่ละ method ควรมีเหตุผลเดียวในการเปลี่ยนแปลงเท่านั้น นักพัฒนาจะไม่ทำงานหรือเปลี่ยนแปลง source code ที่เดียวกันอย่างแน่นอนยกเว้นจะทำงานเดียวกัน หรือ ทำงานด้วยกัน ดังนั้น การออกแบบที่ดีจะช่วยให้ทีมทำงานร่วมกันได้อย่างดี ยิ่งแต่ละส่วนการทำงานเล็กๆแล้ว ก็ยิ่งทำให้คุณภาพของการออกแบบระบบดี รวมทั้งยังลดความเสี่ยงจากการ Merge conflict อีกด้วย

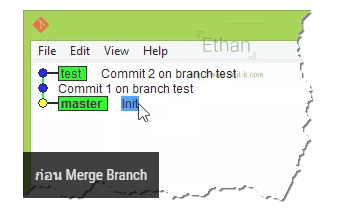
16. Git คืออะไร แตกต่างจาก Github อย่างไร

- Git เป็นระบบที่หน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็ค มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของโปรเจ็คที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้าง ส่วน Github นั้น เป็นเว็บเซิฟเวอร์ที่ให้บริการในการฝากไฟล์ Git มักนิยมใช้ในการเก็บโปรเจ็คแบบ Open Source

17. จุดประสงค์หลักในการ branch คืออะไร

- เป็น feature ที่ช่วยให้นักพัฒนาสามารถที่จะทำงานได้สะดวกขึ้น ยกตัวอย่างเช่น เรามีโค๊ดที่ดีอยู่แล้ว แต่อยากจะทดลองอะไรนิดๆหน่อย หรือแก้ไขอะไรก็ตาม ไม่ให้กระทบกับตัวงานหลัก ก็เพียงแค่สร้าง branch ใหม่ขึ้นมา เมื่อแก้ไขหรือทำอะไรเสร็จแล้ว ก็ค่อยเซฟกลับมาที่ master เหมือนเดิม

18. Fast forward merge ตืออะไรและทำไมการ push ไปที่ remote repo จึงควรจะต้อง merge แบบนี้



* หากทำการ Merge Branch testไปยัง master จะเป็นการ Merge แบบ Fast forward จะเห็นได้ว่า Branch master ไปอยู่ที่ commit เดียวกับ Branch test และสายของ Commit จะเป็นเส้นตรง ดูสวยงาม



เพื่อที่จะส่ง commit ไปยัง remote repo ได้รวดเร็วและสวยงาม

19. หน้าที่หลักของคำสั่ง git pull คืออะไร

- ใช้ดึงความเปลี่ยนแปลงจาก remote มายัง local และรวมเข้าด้วยกัน (มีค่าเท่ากับ fetch+merge)

20.