

11 VERSION CONTROL

เป็นระบบที่จัดการเก็บการเปลี่ยนแปลงของไฟล์ หรือข้อมูล สามารถย้อนกลับไปดูไฟล์บางไฟล์ได้ เพราะมีการเก็บข้อมูล version ก่อนหน้าไว้ด้วย

12 Distributed version control ข้อได้เปรียบ

สามารถแยก version control ออกเป็น repo ย่อยๆได้ แต่ละrepo ย่อยก็สามารถทำ versioning ของตัวเองได้อีกที่

13 Centralized version control ข้อได้เปรียบ

Version ของทุกคนในทีมจะตรงกันทั้งหมดไม่มี local repo

14 แก้ไขปัญหา merge conflict

14-1 ออกแบบงานให้ชัดเจน

14-2 จัดสถานที่ทำงานเป็นกลุ่มๆ

14-3 ควรติดต่อสื่อสารกันให้เข้าใจ

15 ลดปัญหา merge conflict

15-1 ทำการ Merge บ่อยๆ

15-2 ออกแบบระบบงานแบบ SOLID

15-3 พูดคุยสื่อสารกันในทีม

15-4 Mob programming คือการนั่งทำงานร่วมกันทั้งทีม

16 GIT และ Github

Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดการเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจกต์เรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชันต่างๆของโปรเจกต์ที่ใด เวลาใดก็ได้

Github คือ เว็บไซต์เวอร์ที่ให้บริการในการฝากไฟล์ Git (ทั่วโลกมักนิยมใช้ในการเก็บโปรเจกต์ Open Source ต่างๆ ที่ต่างๆ ไม่ว่าจะเป็น Bootstrap, Rails, Node.js, Angular เป็นต้น)

17. branch

Copy โค้ดที่อยู่ภายใน **master** ทั้งหมดไปเป็นอีกโฟลเดอร์หนึ่งแล้วตั้งชื่อใหม่ ทุกคนที่ร่วมกันทำงานก็ให้ตกลงกันว่า จะ **commit, push** โค้ดลงไปที่นี่ ทดสอบโค้ดกันที่นี่ พอหลังจากพัฒนาโค้ดจนดี ไม่มีบั๊ก อยากจะขึ้นโปรดักชันแล้ว ค่อย **merge** เขาเรียกว่าการ **merge** โค้ดกลับเข้าไปยัง **master**

18

19 GIT PULL

ก็อรรวมโค้ดจาก **remote** มายัง **local** โดยที่เราไม่สามารถรู้ได้เลยว่าจะรวมโค้ดอะไรบ้าง รู้แค่หลังจาก **pull** เสร็จ แล้วนั่นเอง