1.ไม่มีความเห็นครับ เพราะแต่ละคน แต่ละงาน รวมถึงลูกค้า ย่อมมีการทำงานที่แตกต่างกันตามความถนัดของผู้พัฒนาซอฟแวร์นั้นๆ

2.ผมว่าประเด็น ตัวไหนดีกว่า ตัวไหนเจ๋งกว่า ผมว่าอยู่ที่ปัจจัยหลายอย่าง วิธีการนำไปใช้ จำนวนคนในทีม ปริมาณงานที่ต้องแก้ไขร่วมกัน และอื่นๆ สิ่งสำคัญคือไม่ว่าจะใช้ตัวไหน และสามารถทำให้ผลงานออกมามีคุณภาพน่าจะเพียงพอ

3. -git branch feature1  
 -git checkout feature1  
 -git commit –m “new branch”  
 -git remote add origin

4.

5.**abcde**

6.

7. MVC คือ สถาปัตยกรรมซอฟต์แวร์ชนิดหนึ่ง ซึ่งปัจจุบันมี Frameworks สำหรับสร้าง Web Apps จำนวนมาก แต่เกือบทั้งหมดมีโครงสร้างแบบ MVC ซึ่งแต่ละ Web Apps จะถูกแบ่งออกเป็นสามส่วน คือ

* Model คือ ส่วนที่ใช้ในการติดต่อกับฐานข้อมูล ซึ่งทำหน้าที่ในการดึงข้อมูลขึ้นมาจาก

ฐานข้อมูลขึ้นมา และทำการจัดการข้อมูลต่างๆไว้ในรูปแบบที่เหมาะสม

* View คือส่วนที่จะนำข้อมูลจาก Model ไปใช้แสดงผลให้ผู้ใช้เห็นผลลัพธ์ออกมา

ใน User Interface

* Controller คือส่วนที่จะคอยรับ Input จาก Client เข้ามาแล้วนำคำสั่งไประมวลผล

เพื่อสั่งงาน View และ Model ให้ประมวลผลออกมาเป็นอย่างไร

**จากภาพ**

**1. เริ่มจาก users ส่ง Request ไปที่ Web App (index) ซึ่งจะถูกส่งต่อให้ Controller**

**ทำการตรวจสอบข้อมูลที่มาให้ (Request Method, Request Parameters)**

**2. แล้ว Controller จะเรียก Method ให้ทำงานเพื่อจัดการ Request นั้น**

**3. Model จะทำการคำนวณและอาจติดต่อกับ Database เพื่อจัดการกับ Request**

**นั้น แล้วส่งผลลัพธ์กลับไปที่ Controller**

**4. เมื่อ Controller ได้ผลลัพธ์จาก Model แล้วก็ใช้ผลลัพธ์นั้นส่งต่อให้ View ทำงาน**

**5. View จะสร้าง Page สำหรับแสดงผลลัพธ์นั้น แล้วส่ง page กลับไปที่ Controller**

**6. Controller ส่ง Page นั้น (เป็น Response) กลับไปยัง Client**

8 Rails เป็น Web Framework มีลักษณะเป็น MVC (Model-View-Controller) พัฒนาโดย 37signals

Rails ถูกออกแบบมาให้มีการใช้งานที่ง่ายและรวดเร็ว ลดปัญหางานทีี่ต้องทำซ้ำๆ ทำให้ได้ productivity

ที่สูงขึ้น สอดคล้องกับ Methodology แบบ Ajile และมีแนวความคิดพื้นฐานคือDon't Repeat Yourself (DRY) อะไรที่อยู่แล้ว ก็ไม่มีความจำเป็นต้องทำสิ่งนั้นซ้ำ Reuseable นั้นเองConvention over configuration คือข้อตกลงที่ Rails สร้างขึ้นเป็นแบบแผน ที่กำหนดมา ขอเพียงเราทำตามรูปแบบเหล่านั้นผลลัพธ์จะออกมาตามที่เราต้องการ โดยที่ไม่จำเป็นต้องใช้การ configuration สำหรับคนที่ใช้ Convention ได้อย่างคล่องแคล่วและถูกต้องแล้วจะทำให้การพัฒนาระบบได้เร็วขึ้นเพราะไม่ต้องมาทำการ configuration ให้ยืดยาว แต่มีข้อดีก็ต้องมีข้อเสียครับ เพราะ Convention ถ้าเป็นมือใหม่หัดเขียนคงต้องศึกษาและทำความเข้าใจกันมากซักหน่อย

9. Herok เป็น Platform as a Service (Paas) ที่ให้เราใช้งานได้ฟรี (มีแบบเสียเงินด้วย) โดยรองรับภาษาโปรแกรมที่หลากหลาย เช่น Ruby, PHP, Node.js, Python, Java, Clojure, Scala และยังสามารถสร้าง buildpack สำหรับภาษาอื่นๆได้ เช่น Lua ที่รันอยู่บน OpenResty ได้อีกด้วย ด้วยเหตุผลที่มันฟรี และ สามารถใช้ Node.js ได้ ( รวมถึงมี add-on สําหรับ database หรืออื่นๆ ให้เราเลย เช่น MongoHQ ทําให้เราไม่ต้องเสียเวลากับเรื่องนี้ และไปใส่ใจอย่างอื่นได้เต็มที่) ในส่วนของคลาสเรียนเราใช้ ภาษา Ruby

10.เพื่อให้นิสิตมีความรู้ หลักการ รวมถึงกระบวนการ พัฒนาซอฟแวร์ เพื่อนำความรู้มาใช้ให้เกิดประโยช์ ทั้งทางตรงและทางอ้อม และเพื่อให้นำไปใช้ทพงานหลังสำเร็จการศึกษาครับ