

1. จงวิจารณ์ข้อต่อไปนี

ไม่เห็นด้วย เพราะ การทำ agile ได้ทำการแบ่งงานให้เป็นงานเล็กๆ และมีการวางแผนในช่วงสั้นๆ โดยได้แบ่งการทำงานออกเป็นช่วงสั้นๆ ในแต่ละช่วงการทำงานจะใช้รูปแบบวงจรการพัฒนาซอฟต์แวร์เต็มขั้น ซึ่งรวมถึงการวางแผน/การเก็บข้อมูลความต้องการ/การออกแบบซอฟต์แวร์/การพัฒนา/และการทดสอบซอฟต์แวร์และในเมื่อหมดช่วงการทำงานนั้นๆ ทีมผู้พัฒนาจะแสดงผลงานที่พัฒนาในช่วงสั้นๆ ให้ลูกค้าได้รับทราบ

2. จงวิจารณ์ข้อต่อไปนี

ไม่เห็นด้วย เพราะ แล้วแต่คนถนัดการใช้งานแบบไหนมากกว่า เช่น

SVN จะเป็นการ checkout โปรเจกต์ลงมา ซึ่งหมายถึงว่า source code ที่เป็นเวอร์ชันล่าสุดจะถูกดึงลงมาไว้ที่เครื่องเรา

Git จะเป็นการ clone คือ การดึงมาทั้ง repository ไม่ว่าจะมิกกิ่ง้านสาขาอะไรก็ตาม ทุกอย่างที่อยู่บน repository จะถูกเอาลงมาทั้งหมด

ข้อ 3. Git branch feature1

Git checkout feature1

Git commit -m “ใส่รายละเอียด”

Git push -u origin master (remote ตั้งไว้ถูกต้องแล้ว)

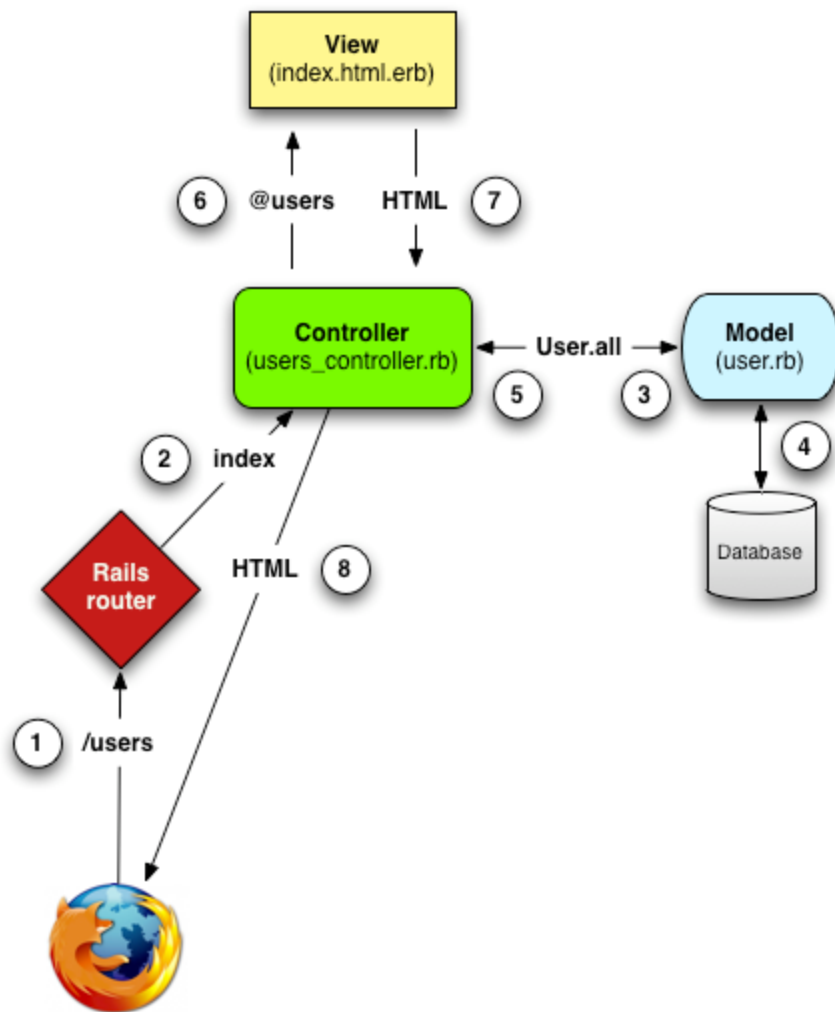
4. ไม่เห็นด้วย เพราะ conflict โดยปกติแล้ว git merge จะรวมโค้ดให้เราเองอัตโนมัติ แต่ก็จะมีข้อยกเว้นเมื่อแก้ไขไฟล์เดียวกัน ลองนึกถึงกรณีที่เราและเพื่อนร่วมทีม แก้ไขไฟล์เดียวกัน Git จะเกิดการ conflict เมื่อเราจะ merge โค้ด โดยไม่รู้ว่าจะใช้โค้ดของเราหรือของเพื่อน วิธีแก้ก็คือ ทำการ edit แล้ว commit ไปใหม่นั้นเอง

5.ผลลัพธ์ที่ได้ คือ abcde

6.ไม่เห็นด้วยกับคำว่า Web Application ที่ทำตามแพชชั่นเพราะ Application

มีส่วนช่วยในชีวิตประจำวันเป็นอย่างมากมีการเข้าถึงได้ง่ายจากเครื่องใดก็ได้ทำให้สะดวกสบายกับการที่ต้องพกพาอุปกรณ์ที่มีการติดตั้งตลอดเวลา ช่วยให้ชีวิตของเราสบายขึ้น

7.



เมื่อมีปฏิสัมพันธ์กับโปรแกรม Rails, เบราร์เซอร์ส่งคำขอซึ่งเป็นที่ได้รับจากเว็บเซิร์ฟเวอร์และส่งผ่านไปยัง รางควบคุมซึ่งเป็นค่าใช้จ่ายในสิ่งที่ต้องทำต่อไป ในบางกรณีการควบคุมทันทีจะทำให้มุมมองซึ่งเป็นแม่แบบที่ได้รับการแปลงเป็น HTML และส่งกลับไปยังเบราร์เซอร์ มากกว่าปกติสำหรับเว็บไซต์แบบไดนามิกควบคุมโต้ตอบกับ รูปแบบซึ่งเป็นวัตถุที่บิคมที่แสดงถึงองค์ประกอบของเว็บไซต์ (เช่น User) และอยู่ในความดูแลของการสื่อสารกับ ฐานข้อมูล หลังจากอัญเชิญรูปแบบการควบคุมแล้วทำให้มุมมองและส่งกลับหน้าเว็บที่สมบูรณ์ให้กับเบราร์เซอร์ เป็น HTML

8. ข้อเสียของการไม่ใช่ Framework

- การจัดการเรื่องระบบ Template ที่ต้องการเปลี่ยน Title ของ Web Application ตามเนื้อหา และหน้าตาของระบบทั้งหมด หากมีการเปลี่ยนแปลงต้องไล่แก้ Code กันให้วุ่น
- ระบบการจัดการกับข้อมูลต้องมาเขียน Class ต่างๆ ในการจัดการ หรือถ้าตอนนี้ดีหน่อยที่มี PDO มาช่วย
- เราเขียน Code ไปได้สำเร็จระบบสวยงามทำงานได้ แต่พอเอาให้คนอื่นพัฒนาต่อ แะๆ เขาจะบอกว่า "ผมเขียนใหม่ดีกว่า"
- หากมีการเขียนกันหลายคนต้องมานั่งวางระเบียบและธรรมเนียมให้ตรงกัน ไม่งั้นเพื่อนๆ จะบอกว่าเอะ "อันนี้คืออะไร มันมาจากไหน"

ข้อดีของการไม่ใช่ Framework

- เขียนโปรแกรมได้ตามใจ ต้องการอะไร เขียนขึ้นมาได้เลย
- ทำงานได้คนเดียวสบายใจ

ข้อดี Rails

- จัดเป็น Web Application Framework กล่าวคือเป็นโครงสร้างสำหรับพัฒนา Web Application ประกอบด้วยชุดเครื่องมือต่างๆ สำหรับอำนวยความสะดวกให้แก่วิศวกรซอฟต์แวร์ ข้อดีของการใช้ Ruby on Rails มีหลายประการคือ
- ไม่ต้องเสียเวลาพัฒนาเครื่องมือเพื่อใช้เอง จึงสามารถใช้เวลาไปกับการสร้างซอฟต์แวร์ตาม ข้อกำหนดได้มากขึ้น
- ส่งเสริมให้วิศวกรซอฟต์แวร์มีระเบียบวินัยที่ชัดเจน เป็นมาตรฐานเดียวกัน
- มีระบบการทดสอบด้วยซอฟต์แวร์ ทำให้การทดสอบครบถ้วนสม่ำเสมอ

ข้อเสีย Rails

- การติดตั้ง Ruby on Rails บนเครื่อง development ยุ่งยากกว่าเครื่องมืออื่นๆ เช่น PHP, Java หรือ .NET
- การ Deploy Rails application ขึ้นเซิร์ฟเวอร์จริง ก็ยิ่งยุ่งยากมากกว่าการติดตั้งบนเครื่อง development

9. Heroku เป็น Platform as a Service (Paas) ที่ให้เราใช้งานได้ฟรี และมีแบบเสียค่าใช้จ่ายโดยรองรับภาษาโปรแกรมที่หลากหลาย เช่น Ruby, PHP, Node.js, Python, Java, Clojure, Scala และยังสามารถสร้าง buildpack สำหรับภาษาอื่นๆได้ เช่น Lua ที่รันอยู่บน OpenResty ได้อีกด้วย

เหมาะกับทุกคน เช่น นักศึกษาอยากลองเขียนเว็บด้วย PHP แต่ไม่ได้เช่า Hosting ก็สามารถใช้ Heroku ได้ หรือแม้แต่บริษัท Start up ที่ไม่ยักวางเครื่องเอง คอนฟิเกอเอง ก็ใช้ได้ เพราะมันสามารถ scale ให้รองรับผู้เยี่ยมชมได้โดยง่ายนอกจากรองรับภาษาโปรแกรมที่หลากหลายแล้ว ตัว Heroku มี App Store ของมันด้วย เรียกว่า [add-ons](#) สำหรับเพิ่มเติมบริการอื่นๆเข้าไปในแอปของเรา เช่น PostgreSQL, MongoDB, Redis เป็นต้น ซึ่งก็มีทั้งฟรี และไม่ฟรีให้เลือกใช้งาน



10. ทำไมต้องบรรจุวิชานี้เข้ามาในหลักสูตร 887342 กระบวนการพัฒนาซอฟต์แวร์

เพื่อให้บัณฑิตสามารถนำความรู้ที่ได้จากวิชากระบวนการพัฒนาซอฟต์แวร์ ไปประยุกต์และใช้งานต่อไปได้ในอนาคตซึ่งเป็นความรู้เบื้องต้นเกี่ยวกับวิศวกรรมซอฟต์แวร์ กระบวนการพัฒนาซอฟต์แวร์ การบริหารโครงการซอฟต์แวร์ กระบวนการวิศวกรรมความต้องการ แบบจำลองระบบ การออกแบบ การสร้างซอฟต์แวร์ การทดสอบ การตรวจสอบความถูกต้อง ตัวชี้วัดซอฟต์แวร์ การประกันคุณภาพซอฟต์แวร์ การจัดการและควบคุมการเปลี่ยนแปลงใน การพัฒนางานด้านซอฟต์แวร์ การบำรุงรักษาซอฟต์แวร์ และนำความรู้ทางด้านต่าง ๆ ที่ศึกษามานำไปใช้ให้เกิดประโยชน์สูงสุด