1.จากคำกล่าวอ้างนั้น ผมไม่เห็นด้วยหากจะกล่าวว่า agile เป็นการออกแบบที่ไม่แน่นอน เพราะมันไม่ถูกเสมอไป หากให้เรียกจริงๆ agile นั้นเปรียบเสมือนกับ “การพัฒนาแบบปรับเปลี่ยนได้” เป็นความจริงที่ว่า agile นั้น ไม่มีการวางแบบแผนที่ตายตัวเหมือนอย่าง Waterfall ซึ่งในตรงส่วนนั้นมันก็มีข้อดีในตัวของมันเองเช่นกัน เพราะหากงานที่ได้รับมาเป็นงานที่ลูกค้ามีความเข้าใจยาก และมีแนวโน้มที่จะเปลี่ยนแปลงระบบตลอดเวลา การทำแบบ agile จึงง่ายต่อการทำงานในรูปแบบนี้ มากกว่าแบบ Waterfall

2.จากคำกล่างอ้างนี้ ผมไม่เห็นด้วยเพราะว่าการทำงาน ของ GIT และ GITHUB กับ CVS หรือ SVN นั้น จะมีการทำงานที่แตกต่างกันออกไป ซึ่งมันก็จะเด่นในแต่ละด้านของมันเอง ซึ่ง git เป็นแบบ distributed version control system ที่สามารถนำมาแยกออกเป็น repo ย่อยๆ ได้ และ แต่ละ repo ย่อยๆ นั้นก็สามารถ ทำ versioning ของตัวเองได้อีกที่ ( จะเป็น local repo ที่ไม่เกี่ยวกับ remote repo ) ส่วนทารง SVN นั้นเป็นแบบ centralized version control system ซึ่งจะบอกได้ว่า version ทั้งหมดนั้นจะถูกเก็บอยู่ที่ remote repo ดังนั้น version ของทุกๆคนที่อยู่ในทีมนนั้นจะตรงกันหมด ( ไม่มี local repo )

3. เริ่มจาก

1. $ git checkout –b feature1

2. $ git add –A

3. $ git commit –m “my branch”

4. $ git push origin feature1

4.

เป้นความจริงที่ว่า การ merge ในแต่ละครั้งนั้นโอกาสที่จะเกิด conflict นั้นสูงมาก แต่วิธีแก้นั้นก็มีอยู่เช่นกัน นั้นคือการคุยกับในทีม หรือก็คือการมี team work นั้นเอง ควรที่จะมีการคุยกันบ่อยๆ และควรมีการแก้ไข conflict ที่เกิดขึ้น หากเป็นงานโปรเจ็คที่ใหญ่ๆ ก็ค่อยแยกออกเป็น commit ย่อยๆ (เพราะถ้าหากเกิด conflict ระหว่างนี้ก็สามารถที่จะทำ Solve ได้)

5. answer : abcde

6.จากคำวิจารณ์ที่กล่าวมานั้น จริงที่ว่า การเอา Software มาลงที่เครื่องเพื่อใช้ส่วนตัวนั้นดี แต่ผมไม่เห็นด้วยที่ ผู้วิจารณ์กล่าวว่าใช้ตามแฟชั่น เนื่องจาก ในสมัยนี้นั้นทุกอย่างทันสมัยและสามารถทำงานได้แม้ว่าจะอยู่ข้างนอกก็ตาม ซึ่งการทำ web application นั้น จะช่วยทำให้รวดเร็วในการใช้งานไม่ต้องเสียเวลาในการลงโปรแกรม และประหยัดเนื้อที่การใช้ข้อมูลของเครื่อง

7.

1 /user ทำการส่งคำขอไป ที่ rails router

2 rail router เรียก index จาก controller

3 controller ทำการเรียก ข้อมูลจาก Model

4 Model ดึงข้อมูลจาก database

5 Model ทำการส่งข้อมูล (user.all) กลับมาให้ที่ controller

6 controller ทำการเรียก view

7 view ส่งกลับมาเป็น HTML

8 controller แสดง view ให้ /user ซึ่งก็คือการแสดงผลทางจอภาพ

8. จากที่ผมได้ เรียน Application mobile มา จึงได้ศึกษาเรื่อง ของ ionic framework ซึ่งในส่วนของ ionic framework นั้น จะเน้นไป ที่การใช้ภาษา JavaScript เป็นหลัก เพราะว่า ionic framework นั้นใช้ Angular JS เป็นฐานในการพัฒนา แต่ถ้าเทียบกับ Rails แล้ว ส่วนของ rails นั้นจะใช้ภาษา Ruby และความรู้เกียวกับ CSS+MySql+HTML เพื่อสามารถใช้งานได้ง่ายมากที่สุด ในส่วนของข้อดีนั้น ทางด้าน ionic นั้นจะ ดีตรงที่มีการเขียนต่างๆ คล้ายกับการเขียน web เพราะใช้ภาษาJavaScript เป็นพื้นฐานซึ่งคนที่เคยเขียนเว็บมาก่อนจะสามารถเข้าใจได้ง่าย แต่ข้อเสียของมันก็คือ การต่อ service ออกไปข้างนอกนั้นจะยาก และต้องมีการปรับการตั้งค่าที่มาก ซึ่งหากเทียบกับ Rails แล้วนนั้นจะถูกออกแบบมาให้เข้าใจง่าย , เขียนง่าย และมีการปรับตั้งค่าที่น้อยกว่า ionic framework เพราะการตั้งค่าต่างๆ จะสามารถตั้งได้ที่โดยทำผ่านระบบ RubyGems ซึ่ง เป็นระบบการจัดการส่วนประกอบในภาษา Ruby

9.เนื่องจากในปัจจุบัน web application เป็นสิ่งสำคัญ ซึ่งหากเราต้องการที่จะพัฒนาหรือศึกษา web application แต่ยังไม่อยากที่จะเช่า Hosting หรือ ไม่อยากตั้งค่า server เอง เพราะไม่มีทรัพยากรมากพอ ก็สามารถใช้ Heroku ได้ เพราะ Heroku นั้นเป็น platform as a service (Paas) ที่สามารถให้เราไปใช้บริการได้ฟรี และสามารถรองรับภาษาที่หลากหลาย ยกตัวอย่างเช่น Ruby , PHP , Node.js , Phyton , Java , Clojure , Scala และ Heroku ยังสามารถ สร้าง buildpack สำหรับภาษาอื่นๆได้ ด้วย

10. เพราะว่าในปัจจุบันการพัฒนา application เป็นไปอย่างรวดเร็วเราจึงต้องควรที่จะรู้การทำงานพื้นฐานต่างๆ ของนักพัฒนาเพื่อ ที่จะทำให้เราทำงานได้มีประสิทธิภาพ ยกตัวอย่างเช่น การทำงานที่มีแบบแผน ที่แน่นอน ควรจะใช้การทำงาน แบบ waterfall แต่หากเป็นการทำงานที่มีการปรับเปลี่ยน หรือ เปลี่ยนแปลงตลอดเวลา ก็ ควรจะปรับไป ใช้ แบบ Agile เป็นต้น และมีการประยุกต์ใช้โปรแกรม หรือ framework ต่างๆ ให้เหมาะกับการทำงานเพื่อที่จะให้งานที่ออกมานั้นมีประสิทธิภาพมากขึ้น