**final\_part\_1**

1.) ไม่เห็นด้วยเพราะว่า Agile เป็นวิธีในการพัฒนาซอฟท์แวร์ โดยมีหลักการในการพัฒนาระบบที่เน้นการทำงานที่รวดเร็ว มีการเตรียมพร้อมที่จะตอบสนองต่อความเปลี่ยนแปลงที่อาจเกิดขึ้นได้ตลอดเวลา โดยเฉพาะความต้องการของผู้ใช้ วิธีพัฒนาแบบนี้ถือเป็นการพัฒนาแบบทำซ้ำ ที่จะต้องมีการพบปะสนทนากับผู้ใช้อยู่ตลอดเวลา และในขณะที่พบปะกันนั้นก็จะถือเป็นช่วงระยะเวลาของการส่งงานไปในตัวด้วย การส่งงานแต่ละครั้งของวิธีการพัฒนาแบบนี้ โดยส่วนใหญ่จะเป็นการพัฒนาเฉพาะส่วนย่อย ๆ แล้วค่อยทยอยส่งให้กับผู้ใช้ เมื่อผู้ใช้ได้ทดสอบหรือประเมินระบบแล้ว ถ้าต้องการปรับเปลี่ยนตรงส่วนใดก็สามารถทำได้โดยที่ไม่ต้องรื้อระบบใหม่ทั้งหมดสามารถแก้ได้ตรงจุดตามที่ผู้ใช้ต้องการได้เลย ต่างจาก Waterfall ปัญหาสำคัญของ Waterfall คือขั้นตอนของการพัฒนาที่ไม่ยืดหยุ่น เพราะตัวงานจะแบ่งเป็นช่วงๆแบบตายตัว ทำให้มีข้อผูกมัดตั้งแต่เริ่มโครงงานและไม่สามารปรับเปลี่ยนความต้องการผู้ใช้ได้ หมายความว่าการพัฒนาโดยใช้ Waterfall นั้น ไม่เหมาะกับงานที่ความต้องการของผู้ใช้เข้าใจยาก และมีแนวโน้มว่าจะเปลี่ยนแปลงตลอดเวลา

2.) เห็นด้วยเพราะว่า เช่น เวลาเราเขียนโค้ดหรือต้องการเก็บไฟล์​ บางทีเราอาจต้องการติดตามการเปลี่ยนแปลงของไฟล์ เช่นสำหรับ text file เราอยากจะเก็บความเปลี่ยนแปลงของ text ที่เราแก้ และกลับมาดูเวอร์ชันที่เราเก็บไว้ได้ เช่นเวอร์ชัน 1, 2, 3, และเราต้องการเปรียบเทียบความแตกต่างของไฟล์เวอร์ชันต่างๆได้ เราอยากกลับไปดูได้ว่ามีอะไรเปลี่ยนแปลงไปบ้างในแต่ละเวอร์ชันของไฟล์ โดยวิธีการจัดเก็บไฟล์แบบนี้เรียกง่ายๆ ว่า Version Control System (VCS) นั่นเอง และที่คิดว่า distributed version control ดีกว่า centralized version control เพราะ ถ้าสมมติว่า central repo เกิดพังขึ้นมาหรือผู้ใช้ไม่สามารถติดต่อกับ central repo ได้ก็จะทำให้ไม่สามารถดึงไฟล์ที่พึ่งเปลี่ยนแปลงมาทีตัวเองได้และ distributed version control จึงเร็วกว่า และผู้ใช้งานก็สามารถแก้ได้ทุกที่ โดยไม่จำเป็นจะต้องติดต่อกับ central repo ก็ได้

3.) ใช้คำสั่ง

git branch feature

git add –A

git commit –m “ ”

git push –u origin feature

4.) เราก็รับฟังไว้แล้วก็นำมาคิด คือเราต้องเข้าใจก่อนว่า merge conflict เกิดขึ้นมาจากอะไร ปัญหาใหญ่ ๆ ของ Merge conflict เกิดจากจำนวน source code ที่ชนหรือขัดแย้งกันมากเกินไป แล้วเราก็ต้องมาหาวิธีแก้ไข เช่น ทำการ Merge บ่อย ๆ นั่นคือ ทุกครั้งเมื่อคุณทำการเปลี่ยนแปลง หรือ commit source code นั่นเองจะช่วยลดข้อขัดแย้งต่าง ๆ ลงไปอย่างมากถึงจะเกิดข้อขัดแย้ง ก็เป็นเพียงปัญหาเล็ก ๆ ซึ่งสามารถแก้ไขได้อย่างง่ายดาย หรือมีอีกหลายวิธี ดังนั้นไม่ว่าจะไปทำที่ไหนเราก็จะไม่ต้องกลัวการ conflict อีกต่อไป

5.) ผลลัพธ์ที่ได้คือ abcde"a".."e"

6.) เรื่องนี้ก็แล้วแต่บุคคลค่ะ บางคนที่ชอบแบบ web application ก็อาจจะเป็นเพราะว่า ชอบอะไรที่มันง่ายๆ สะดวกสบาย และรวดเร็ว ไม่ต้องทำอะไรมากก็แค่โหลดมาติดตั้งสักครู่ก็สามารถใช้งานได้เลย ส่วนคนที่บอกว่าอยากได้แบบ CD แล้วเอามาลงเองในเครื่องเพื่อใช้นั่นก็อาจจะเป็นเพราะเขาคุ้นเคยกับแบบนี้ หรือที่ไม่ชอบแบบ web application เพราะว่าเขากลัวว่าจะโดนไวรัสก็ได้ค่ะ เพราะว่าเราโหลดอะไรมาก็ไม่มีอะไรยืนยันว่ามันปลอดภัยร้อยเปอร์เซ็น ทางที่ดีเขาจึงเลือกแบบซื้อ CD มาแล้วลงเครื่องเองแบบนี้มั่นใจกว่าและก็สามารถใช้งานได้เหมือนกัน

7.) กลไกของ MVC ใน Rails

1. เริ่มจาก Client ส่ง user ไปที่ Rails router ทำการตรวจสอบข้อมูลที่มาให้ (user)
2. แล้ว Rails router จะเรียก Controller โดยส่งผ่าน index เพื่อจัดการ Request นั้น
3. Controller ส่ง @users ไปที่ View และ View จะส่ง HTML กลับมาให้ Controller
4. Controller จะส่ง User.all ต่อให้ Model
5. Model จะทำการคำนวณและอาจติดต่อกับ Database เพื่อจัดการกับ Request นั้น แล้วส่งผลลัพธ์กลับไปที่ Controller
6. Controller จะส่ง HTML (เป็น Response) กลับไปยัง Client

8.) ข้อดีของ Rails

* Code เขียนง่าย ไม่ซับซ้อน เมื่อเทียบกับ java
* ถ้า Code สั้น ดังนั้นโอกาสในการเกิด Bug จึงน้อย
* เป็น Open source
* มี Library ให้นำไปใช้เป็นจำนวนมาก
* ง่ายต่อการนำไปขยายและพัฒนาต่อ (extend)
* เป็น pure Object Oriented

ข้อเสียของ Framework คือ

* มีขนาดของ source โดยรวมใหญ่กว่าที่ควรจะเป็น เพราะบางอย่างเราไม่จำเป็นต้องใช้ แต่ว่ามันดันมีใน Framework
* อาจจะทำให้ผู้ไม่ประสงค์ดีโจมตีได้ง่ายขึ้น เพราะว่า โครงสร้างของ Framework ใคร ๆ ก็รู้ (ถ้าศึกษา)

9.) Heroku คือผู้ให้บริการ Platform as a Services (PaaS) เช่น แอพพลิเคชันเซิร์ฟเวอร์, ดาต้าเบสเซิร์ฟเวอร์ หรือมิดเดิลแวร์อื่นๆ ไม่ต้องเสียเวลาหา software ไม่ต้องหา server และลดความยุ่งยากในการ configuration เพราะเพียงแค่คลิกเลือกภาษาที่ต้องการสร้าง app ไม่ถึงนาทีเราก็มี environment พร้อมใช้งาน - และมีบทบาทกับการพัฒนา คือ Heroku เป็น PaaS อีกอันหนึ่งที่สามารถใช้ภาษาที่หลากหลายในการพัฒนาได้อาทิเช่น Ruby, Node.js, Clojure, Java, Python, and Scala ซึ่ง Cloud Platform นี้ได้ถูกซื้อไปโดย Salesforce

10.) เพราะ เพื่อให้ได้เข้าใจว่า Software มีความสำคัญเกี่ยวข้องกับชีวิตเรามากขึ้นเรื่อยๆ ไม่ว่าจะเกี่ยวข้องกับวัตถุในชีวิตประจำวันมนุษย์เช่นโทรศัพท์ เครื่องซักผ้า รถยนต์ รถไฟฟ้า หรือเกี่ยวกับธุรกิจเช่นธนาคาร ดังนั้นการสร้าง Software ต้องสร้างให้ได้คุณภาพดี ไว้ใจได้นักศึกษากล้านั่งรถไฟฟ้าที่ตนเองเป็นคนเขียน Software ควบคุมการเบรคหรือไม่ ปัญหาการสร้าง Software ในปัจจุบันยังเป็นวิกฤต คือไม่น่าเชื่อถือ ทำงานไม่ถูกต้องไม่ได้ตามที่คาดไว้ สร้างไม่สำเร็จ ไม่ทันตามกำหนดเวลา งบประมาณ หรือต้องล้มเลิกกลางคันจำนวนมาก