1. เห็นด้วยว่า Waterfall Model  ดีกว่าเพราะ Agile Method ไม่มีอะไรที่เหมือนกับ Waterfall Model เลย การพัฒนาซอฟต์แวร์แบบ Agile เป็นแนวคิดใหม่สำหรับการพัฒนาซอฟต์แวร์ที่พยายามจะแทรกตัวเข้าไปในวิธีการแบบเดิม เพื่อให้งานสั้นลง ประหยัดเวลา และประหยัดงบประมาณ  การพัฒนาซอฟต์แวร์แบบ Agile มีหลายวิธีซึ่งส่วนใหญ่แล้วจะเป็นการพัฒนาในช่วงเวลาสั้นๆ ซึ่งอาจจะใช้เวลาประมาณ 1-4 สัปดาห์ แต่ Waterfall Model เป็นวิธีการพัฒนาระบบที่ใช้กันอยู่ค่อนข้างแพร่หลาย ซึ่งวิธีนี้จะมีลำดับขั้นตอนที่ตายตัว เริ่มตั้งแต่รวบรวมข้อมูล กำหนดความต้องการของผู้ใช้ วิเคราะห์ทางเลือก ออกแบบ เขียนโปรแกรม ทดสอบระบบ และสุดท้ายทำการติดตั้งระบบ โดยแต่ละส่วนของขั้นตอนดังกล่าวจะถือเป็นตัววัดความก้าวหน้าของงานซึ่งเป็นวิธีการทำงานที่มีการวางแผนอย่างละเอียดดีกว่า Agile Method  อย่างมากเลย
2. เห็นด้วยค่ะว่าในเรื่องการทำ version control นั้น Git และGithub ยังไงก็ต้องดีกว่า CVS หรือ SVN อย่างแน่นอน

ก่อนอื่นต้องบอกความหมายก่อนว่า Version control คือ ระบบที่จัดเก็บการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์หนึ่งหรือหลายไฟล์เพื่อที่สามารถเรียกเวอร์ชั่นใดเวอร์ชั่นหนึ่งกลับมาดูเมื่อไรก็ได้

จะมียุคหนึ่งที่ SVN รุ่งเรือง ถือว่าเป็นจุดเริ่มต้นของ Version Control แบบกว้างขวางของโลก แต่ SVN ก็มีจุดอ่อนมากมายหลายหลาก ยกตัวอย่างเช่นการที่ต้องออนไลน์ถึงจะทำอะไรกับ Repository ได้ เป็นต้น Git จึงถูกคิดค้นขึ้นมาเพื่อทำหน้าที่แบบเดียวกัน แต่ดีกว่าในหลายๆด้านเช่น สามารถทำได้แบบ offline เก็บการเปลี่ยนแปลงของแต่ละไฟล์ไว้อย่างเป็นรูปแบบ ทำให้เราสามารถย้อนการกระทำใดๆได้ เป็นต้น จนตอนนี้กลายเป็น Version Control ที่ได้รับการยกย่องว่าดีที่สุดเป็นที่เรียบร้อย ถึง SVN จะยังมีคนใช้อยู่ แต่ก็มีคนแนะนำให้ย้ายไปใช้ Git กันอยู่เรื่อยๆ

1. git add feature1

git commit -m "finish"

git remote add origin https://github.com/ชื่อgithubที่ต้องการนำงานไปไว้/ชื่อrepoที่ใช้เก็บงาน

git push -u origin master

1. เห็นด้วยค่ะเป็นปัญหาที่ยังไงก็ต้องได้เจอและไม่สามารถหลักเลี่ยงมันมันได้

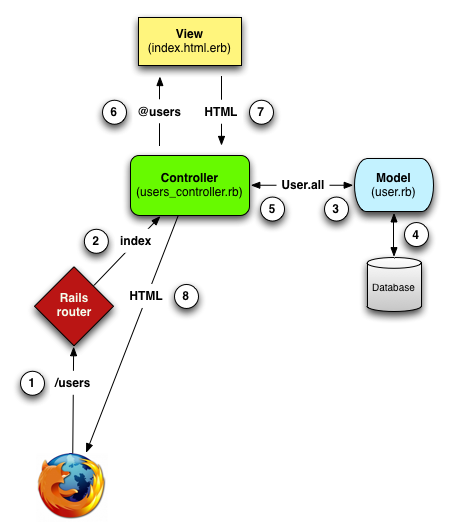
เพราะการmerge conflict คือ เกิดกรณีที่มีการแก้ไขไฟล์เดียวกัน เวลาเดียวกัน Git จะไม่รู้ว่าเราควรจะใช้ไฟล์ไหนดี ถ้าต้องการหลีกเลี่ยงปัญหาเหล่านี้ ให้ทำการ merge บ่อย ๆ ไปเลย นั่นคือ ทุกครั้งที่ทำการเปลี่ยนแปลง หรือ commit source code นั่นเอง จะช่วยลดข้อขัดแย้งต่างๆ ลงไปอย่างมาก ถึงจะเกิดข้อขัดแย้ง ก็เป็นเพียงปัญหาเล็ก ๆ ซึ่งสามารถแก้ไขได้อย่างง่ายดาย

1. abcde"a".."e"
2. ไม่เห็นด้วยเท่าไหร่ค่ะ เพราะการที่จะได้ซอฟต์แวร์มานั้นก็ต้องหาที่เป็นของลิขสิทธิ์และอาจมีค่าใช้จ่ายที่สูง แต่เทคโนโลยี Web Application มีความสะดวกมาก เพราะ Web Application สามารถตอบสนองปัญหาต่างๆได้เป็นอย่างดี และเป็น Client-Server Application ได้เป็นอย่างดี ตัวโปรแกรมของ Web Application จะถูกติดตั้งไว้ที่ Server คอยให้บริการกับ Client และที่ Client ก็ไม่ต้องติดตั้งโปรแกรมเพิ่มเติม สามารถใช้โปรแกรมประเภท Brower ที่ติดมากับ OS ใช้งานได้ทันที อย่าง Internet Explorer หรือโปรแกรมฟรี ได้แก่ FireFox, Google Chrome ซึ่งกำลังเป็นที่นิยมเป็นอย่างมาก ด้วยความสามารถของ Brower ที่หลากหลาย ทำให้ไม่จำกัดว่าเครื่องที่ใช้เป็น OS อะไร หรืออุปกรณ์อะไร อย่างอุปกรณ์ TouchPad หรือ SmartPhone ก็สามารถเรียกใช้งานได้ ลดข้อจำกัดเรื่องสถานที่ใช้งานอีกด้วย
3. **Model** คือ ส่วนที่ใช้ในการติดต่อกับฐานข้อมูล ซึ่งทำหน้าที่ในการดึงข้อมูลขึ้นมาจาก  
   ฐานข้อมูลขึ้นมา และทำการจัดการข้อมูลต่างๆไว้ในรูปแบบที่เหมาะสม

**View** คือส่วนที่จะนำข้อมูลจาก Model ไปใช้แสดงผลให้ผู้ใช้เห็นผลลัพธ์ออกมา

ใน User Interface

**Controller** คือส่วนที่จะคอยรับ Input จาก Client เข้ามาแล้วนำคำสั่งไประมวลผล   
เพื่อสั่งงาน View และ Model ให้ประมวลผลออกมาเป็นอย่างไร



1.เริ่มจาก Client ส่ง Request ไปที่ Rails router

2. Rails router ทำการส่ง Web App ที่ได้จาก client เพื่อส่งต่อไปให้ Controller ทำการตรวจสอบข้อมูล

3.แล้ว Controller จะเรียก Method ให้ทำงานเพื่อจัดการ Request นั้น

4. Model จะทำการคำนวณและอาจติดต่อกับ Database เพื่อจัดการกับ Request

5.ส่งผลลัพธ์กลับไปที่ Controller

6.เมื่อ Controller ได้ผลลัพธ์จาก Model แล้วก็ใช้ผลลัพธ์นั้นส่งต่อให้ View ทำงาน

7. View จะสร้าง Page สำหรับแสดงผลลัพธ์นั้น แล้วส่ง page กลับไปที่ Controller

8. Controller ส่ง Page นั้น (เป็น Response) กลับไปยัง Client

1. framework ที่เคยใช้งานคือ การทำ mvc ในโปรแกรม [Secure Shell](https://th.wikipedia.org/wiki/Secure_Shell) ไว้ในฐานข้อมูล angsila ที่ได้จากมหาวิทลัย ซึ่งสามารถจัดการได้ยากกว่า และไม่สามารถประมวลผลหรือทำการ test ในโปรแกรมดูได้ว่าสิ่งที่ได้ทำหรือแก้ไปนั้นถูกหรือไม่ แต่พอได้มาลองทำ mvc ใน Rails นั้น ทำให้จัดการการทำงานที่ดีและง่ายมากขึ้นเพราะมีข้อดีหลายอย่างที่ตัวโปรแกรม [Secure Shell](https://th.wikipedia.org/wiki/Secure_Shell) นั้นไม่สามารถทำได้
2. Heroku คือผู้ให้บริการ Platform as a Services (PaaS) เช่น แอพพลิเคชันเซิร์ฟเวอร์, ดาต้าเบสเซิร์ฟเวอร์ หรือมิดเดิลแวร์อื่นๆ ไม่ต้องเสียเวลาหา software ไม่ต้องหา server และลดความยุ่งยากในการ configuration เพราะเพียงแค่คลิกเลือกภาษาที่ต้องการสร้าง app ไม่ถึงนาทีเราก็มี environment พร้อมใช้งาน ที่สำคัญฟรี และเราสามารถพัฒนา web application ของเราได้โดยใช้ Heroku เป็นตัวจัดการ
3. เพื่อให้นิสิตมีพื้นฐานความคิดในเรื่องของการพัฒนาระบบว่ามีโปรแกรมหรือซอฟต์แวร์อะไรที่จำเป็นต้องใช้บ้าง ได้เรียนรู้ถึงการวิเคราะห์และออกแบบระบบเป็นและเพื่อให้สามารนความรู้ที่ได้เรียน ได้ศึกษาจากวิชานี้ไปใช้ต่อยอดได้ในอนาคต