1.กระบวนการพัฒนาซอฟแวร์แบบWaterfall และ แบบ Agile ล้วนมีข้อดี ข้อเสียแตกต่างกัน ซึ่งผู้พัฒนาซอฟแวร์จะต้องเลือกแนวทางการใช้ให้เหมาะสมกับรูปแบบงาน ความต้องการหรือเงื่อนไขของการส่งงานของตนเองกับผู้ว่าจ้าง แต่ในความคิดของดิฉัน ดิฉันชอบแบบ Agile มากกว่า เพราะลักษะการกำหนดส่งงานมีความยืดหยุ่น สามารถเปลี่ยนแปลงได้ ได้พบลูกค้า บ่อยๆ ทำให้ได้ทราบความต้องการของลูกค้า

2.ระบบควบคุมเวอร์ชั่น Centralized และ Distributedมีข้อดีและข้อเสียแตกต่างกัน ขึ้นอยู่กับความชอบของทีม แต่ในความคิดเห็นของดิฉัน ดิฉันชอบแนวทางแบบ Distributed มากกว่า เนื่องจากว่า เป็นแนวทางที่มีความยืดหยุ่น ทำงานแบบออฟไลน์ได้ ผู้ใช้สามารถทำงานได้ทุกที่โดยไม่จำเป็นต้องทำงานพร้อมกันในทีม ทุกคนสามารถเปลี่ยนแปลงงานในเครื่องคอมของตนเองก่อนที่จะย้ายลง repo กลางได้ การทำ Branching (แยกงานไปทำคนละสาขา) หรือ Merging (การเอางานมารวมกัน) สามารถทำได้ง่ายและรวดเร็ว

3. git checkout -b feature1

git add -A

git commit -m "Add a feature1"

git push -u origin feature1

4.

- การ merge จะทำ เมื่อเรา branch เพื่อไปทำ feature ต่างๆ และต้องการ merge กลับมา ยัง repo กลาง

- Conflict จะเกิดขึ้น เมื่อโปรแกรมเมอร์ 2 คน มาทำงานไฟล์เดียวกัน และแก้ไขโค้ดให้แตกต่างกัน

- วิธีการแก้ conflict คือ 1.ให้หัวหน้าตัดสินใจ 2. พูดคุยกันในทีม

- การป้องกัน conflict คือการทำ software engineering ที่ดี คือ การแบ่งงานกันทำอย่างชัดเจน

5. ('a'..'e').each {|ch| print ch} ผลลัพธ์ คือ abcde"a".."e"

6.ในความคิดของดิฉัน ดิฉันไม่เห็นด้วยกับแนวคิดนี้ เนื่องจากว่า การทำงานบนเว็บไซต์ จะสามารถ update version ได้ง่ายเพียงแค่ดาว์นโหลดจากเว็บไซต์ อีกทั้งยังสามารถทำได้ทุกที่ที่มีอินเตอรเน็ต แต่การติดตั้งซอฟแวร์ลงเครื่อง เวลาจะ update version เราจะต้องไปหาแผ่น CD เพื่อมา update ทำให้ไม่สะดวกในการใช้งาน และไม่รักษ์สิ่งแวดล้อม

7.1 ผู้ใช้พิมพ์บน URI ว่าต้องการข้อมูล user

7.2 Rail router จะส่งข้อมูลไปที่ controller คือไฟล์ user\_controller.eb

7.3 controller จะส่งข้อมูลไปที่ฝั่ง Model คือไฟล์ user.rb

7.4 Model จะไปดึงข้อมูล user จาก database

7.5 Model ส่งข้อมูลจาก database กับไปให้ controller

7.6 controller ส่งข้อมูล user ไปให้ view เพื่อแสดงผล ซึ่งชื่อไฟล์ index.html.erb

7.7 View ทำการแสดงผลในรูปแบบภาษา HTML

8.ดิฉันเคยใช้งาน Codeigniter framework เพื่อใช้ในการพัฒนาเว็บไซต์ Rails และ Codeigniter มีความคล้ายกันตรงที่มีลักษณะเป็น MVC (Model-View-Controller)

Rails มีข้อดีคือ ถูกออกแบบมาให้มีการใช้งานที่ง่ายและรวดเร็ว ลดปัญหางานที่ต้องทำซ้ำๆ ทำให้ได้ productivity ที่สูงขึ้น สอดคล้องกับ แนวคิดแบบ Agile แต่ Rails มีข้อเสีย คือไม่ค่อยเหมาะจะเอาไปใช้ในโปรเจคท์ใหญ่ๆ

Codeigniter มีข้อดีคือการทำงานพื้นฐานที่ต้องไม่หนักไป, การต่อยอด framework ที่ทำได้ง่าย, การใช้งานที่เข้าใจได้ไม่ยากเกินไป แต่ก็มีข้อเสียไม่มี function ส่วน Javascript ให้เรียกใช้

9. Heroku คือ Cloud Application Platform หรือ Platform as a Services ( PaaS )ซึ่งสามารถใช้งานได้ฟรี สามารถใช้ Node.js ได้ รวมถึงมี add-on สําหรับ database หรืออื่นๆ เช่น MongoHQ

Heroku มีบทบาทสำคัญ คือ สามารถนำโปรเจคที่เราสร้างขึ้นไปใช้งานบนเว็บไซต์โดยไม่ต้องติดตั้งเครื่อง server

10.สาขาวิชาวิทยาการสารสนเทศบรรจุวิชากระบวนการพัฒนาซอฟแวร์เข้ามาในหลักสูตรเพราะต้องการให้นิสิตได้มีความรู้ ความเข้าใจเกี่ยวกับกระบวนการในการพัฒนาซอฟแวร์ เพื่อให้นิสิตสามารถสร้างโปรเจคออกมาให้ตรงตามความต้องการของลูกค้า บริหารจัดการงบการเงินอบ่างพอดี และสามารถส่งงานได้ตรงเวลา โดยยึดแนวทางการใช้ Agile