1.

ผมไม่เห็นด้วยกับความคิดนี้ เพราะว่า การทำแบบ agile นั้นไม่มีขั้นตอนที่แน่นอนก็จริง แต่ใช่ว่างานที่ได้จากการทำแบบ waterfall นั้นจะได้งานที่ดีกว่าแบบ agile

การทำแบบ agile นั้นมีประโยชน์ตรงที่งานสามารถยืดหยุ่นได้ สามารถเปลี่ยนไปตามความต้องการของผู้ใช้งานได้ตลอดเวลาการพัฒนา เพราะผู้พัฒนาจะมีการแลกเปลี่ยนข้อมูลกับผู้ใช้งาน จึงทำให้ได้งานตรงตามความต้องการที่กำหนดไว้ให้ ซึ่งแตกต่างจากแบบ waterfall ที่มีขั้นตอนการทำงานที่แน่นอน แก้ไขความต้องการได้ยาก

2.

ผมมองว่า CVS/SVN ยากที่จะสูญพันธุ์ไป เพราะว่า Git/GitHub กับ CVS/SVN นั้น มีการใช้งานที่แตกต่างกัน คือ Git/GitHub มีการทำงานแบบ distributed ซึ่งสามารถแยกการทำงานเป็น repo ย่อยๆ หรือเรียก local repo ส่วน CVS/SVN นั้นมีการทำงานแบบ centralized ซึ่งแตกต่างจาก Git/GitHub ตรงที่ version ทั้งหมดจะไปรวมอยู่ที่ remote repo

3.

$ git checkout -b feature1

$ git add –A

$ git commit

$ git push origin feature1

4.

อาจจะจริงแบบที่รุ่นพี่ว่า ในการทำ Git ร่วมกันนั้นมีโอกาสค่อนข้างมากที่จะเกิดการ conflict เวลา merge แต่ถ้ากลุ่มเรามีการแบ่งงานกันเป็นส่วน เช่น 1 ไฟล์งานมีการแก้ไขเพียงคนเดียว จะสามารถช่วยลดโอกาสการเกิด conflict ได้

5.

Answer: abcde

6.

Web application นั้นเป็นที่นิยม เพราะว่า ติดตั้งง่ายกว่า software application, ถูกกว่า เพราะ web application บางตัวที่เสียเงินยังถูกกว่า software application เสียอีก รวมไปถึงการ update version ทาง web application นั้นสะดวกสบายกว่า software application เป็นอย่างมาก

7.

1. User เชื่อมต่อไปยัง rails router ซึ่งเป็นไฟล์ที่จัดการว่า URL อะไร จะถูกนำไปใช้บน Controller อะไร

2. Rails router ติดต่อไปยัง controller ว่า user ต้องการดู view ของหน้า index

3. Controller ติดต่อไปยัง model เพื่อขอข้อมูล

4. Model ดึงข้อมูลหน้า index จาก database

5. Model ส่งข้อมูลที่ query จาก database ไปให้ Controller

6. Controller ส่งข้อมูลหน้า index ไปให้ view

7. View ส่งค่า HTML ไปให้ Controller

8. Controller ทำการแสดง View หน้า index ให้ user

8.

Rails framework นั้นมีจุดเด่นที่ DRY คือ Don't repeat yourself ซึ่งเป็นการไม่ทำซ้ำตัวเอง เช่น header footer ต่างๆ จะทำการสร้างเพียงครั้งเดียวพอถ้ามี content เหมือนกัน เพราะ Rails ได้ใช้ Ruby ที่จะนำสิ่งๆ นี้เข้ามา คุณจะเจอการทำสำเนาเล็กๆ ของการเรียกใช้ header หรือ footer จะมีการใช้งาน active record ทำให้มีการประกาศเพียงครั้งเดียว แล้วเรียกใช้ได้ตลอด ซึ่งต่างจาก ionic framework ที่ผมไปศึกษา เพราะจะมีการกำหนด header ทุกครั้งในแต่ละ view

9.

Heroku นั้นเป็น PaaS (Platform as a service) ซึ่งมีบทบาทกับ web application มาก เพราะปัจจุบัน cloud computing กำลังมากแรง มีจุดเด่นตรงที่การปรับเปลี่ยน system feature สามารถเปลี่ยนและอัพเกรดได้รวดเร็ว แถมช่วยลดค่าใช้จ่ายในการดูแลระบบ

10.

ที่ต้องบรรจุวิชานี้เข้ามาในหลักสูตรเพื่อให้นิสิตรู้จักการทำ version control เพราะการออกไปทำงานนอกรั้วมหาลัยนั้น จะต้องทำงานร่วมกับผู้อื่น มีหลักการและมาตรฐาน ไม่อย่างนั้นนิสิตจะไม่สามารถทำงานร่วมกับผู้อื่นได้ ถ้าทำได้งานก็อาจจะเสร็จล่าช้า