1.ไม่เห็นด้วยกับคำกล่าวที่ว่า agile คือเเฮ็กเกอร์อย่างเดียว เพราะเป็นหลักการในการพัฒนา software ที่เน้น Rapid and flexible response to change ทำให้การพัฒนาว่องไวมีการทำเรื่อยๆไม่ต้องหยุด สามารถรองรับความเปลี่ยนแปลงนั้นได้อย่างรวดเร็ว ลูกค้าและผู้พัฒนาได้ทำงานร่วมกันทำให้เมื่อพบข้อผิดพลาด การแก้ไขจะทำได้ง่าย

water fall นั้นอาจมีขั้นตอนที่ชัดเจนก็จริงแต่ลูกค้าจะเห็นและทดลองใช้ software ก็ต่อเมื่อถึงขั้นตอนสุดท้าย หากมีบางอย่างที่ไม่ตรงกับความต้องการของลูกค้า การแก้ไขก็ทำได้ยาก แพง เสียเวลา

2.ในปัจจุบันก็มีการใช้ทั้ง Git และ CVS

การ pull โปรเจคลงมาจาก repository

SVN จะเป็นการดึง source code ที่เป็นเวอร์ชั่นล่าสุดลงมาไว้ที่เครื่องเรา

Git จะเป็นการ clone คือ การดึงมาทั้งหมด ไม่ว่าจะมีกิ่งก้านสาขาอะไรก็ตาม ทุกอย่างที่อยู่บน repository

การ upload file สู่ repository

SVN นั้น จะเรียกว่าการ commit และหลังจากที่ commit ไปแล้วไฟล์เหล่านั้นก็จะอยู่ใน repository ทันที

Git การ commit จะเป็นการเก็บอยู่ใน local ก่อน ซึ่งเราสามารถ commit แต่ละส่วนของโปรแกรมทีละนิดๆ แล้วค่อย push ขึ้นสู่ remote repository ทีเดียวก็ได้

ลำดับของ revision

SVN จะใช้เลขจำนวนเต็ม(Integer) รันไปเรื่อยๆ ยิ่ง commit บ่อยก็จะทำให้เลข revision มีเยอะขึ้นๆ

Git จะใช้ hash ด้วย SHA-1 hash algorithm เนื่องจาก git จะมีการ commit สู่ local repo. ก่อน เพราะถ้าใช้เป็นเลขรันไปเรื่อยๆเหมือน SVN ก็จะทำให้เกิดเลข revision ซ้ำกันได้

การทำงาน Online / Offline

SVN เป็น version control แบบ centralized หมายความว่า ทุกอย่างจะถูกเก็บในตัวกลาง ซึ่งก็คือ remote repo. ทั้งหมด ทำใช้งานจำเป็นต้องเชื่อมต่อ network ไปยัง repo.

Git เป็น version control แบบ distributed คือ หลังจากที่ clone remote repo. มาแล้ว การจะ commit ก็สามารถทำได้แบบ offline เลยเพราะเป็นการ commit ไปที่ local repo. แต่ถ้าเราจะ push ไปไว้ที่ remote repo. แน่นอนว่าต้องเชื่อมต่อ network เช่นเดียวกัน

การที่เราจะเลือกใช้ก็ขึ้นอยู่ที่ปัจจัยต่างๆ เช่น วิธีการนำไปใช้ จำนวนคนในทีม ปริมาณงานที่ต้องแก้ไขร่วมกัน จึงคิดว่าไม่มีอันไหนที่จะสูญพันธ์

3.Rails new feature1

Git commit –m

Git remote add origin

Git push –u origin master

4.ที่เกิด conflict บ่อยๆ ก็ต้องเอามาดูและแก้ไข

5.

6. Web application นั้นไม่ใช่สิ่งที่ทำตามแฟชั่น แต่คือการเขียนโปรแกรมที่ให้ตอบสนองต่อผู้ใช้มากที่สุด จะอยู่ในรูปแบบของเว็บ ซึ่งนั้นก็คือสามารถใช้งานได้ทุกหน้าจอที่มีความแตกต่างของขนาดหน้าจอ สามารถยืดหยุ่นและหดตัวได้ตามสภาพของ UI

7.

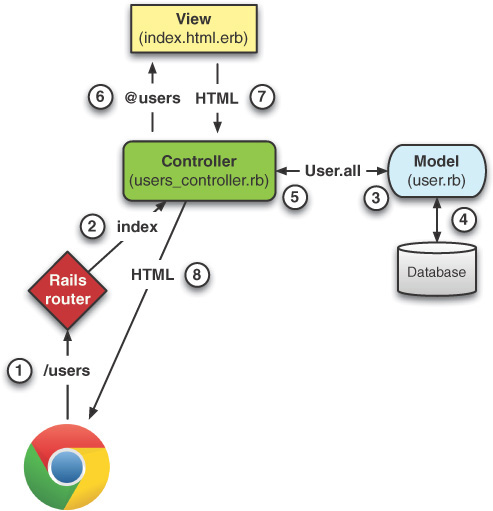


Figure 2.7 A detailed diagram of MVC in Rails.

1.The browser issues a request for the /users URL.

2.Rails routes /users to the index action in the Users controller.

3.The index action asks the User model to retrieve all users (User.all).

4.The User model pulls all the users from the database.

5.The User model returns the list of users to the controller.

6.The controller captures the users in the @users variable, which is passed to the index view.

7.The view uses embedded Ruby to render the page as HTML.

8.The controller passes the HTML back to the browser.

8.framework ที่เคยใช้ คือ Bootstrap ซึ่งมีการอัปเดทอยู่ตลอดเวลา เพื่อรองรับการทำงานใหม่ๆและการแก้ไขปัญหาก็ทำได้เร็ว รวมทั้งเว็บแอพลิเคชัน App บนมือถือ

**ข้อดีของ Framework**

•รวดเร็ว

•ทำงานเป็นทีมได้

•เหมาะกับองค์กรใหญ่ ๆ

**ข้อเสียของ Framework**

•มีขนาดของ source โดยรวมใหญ่กว่าที่ควรจะเป็น เพราะบางอย่างเราไม่จำเป็นต้องใช้ แต่ว่ามันดันมีใน Framework

•อาจจะทำให้ผู้ไม่ประสงค์ดีโจมตีได้ง่ายขึ้น เพราะว่า โครงสร้างของ Framework ใคร ๆ ก็รู้ (ถ้าศึกษา)

**ข้อดีของ Rails**

•Code เขียนง่าย ไม่ซับซ้อน เมื่อเทียบกับ java

•Code สั้น ดังนั้นโอกาสการเกิด Bug จึงน้อย การทำงานเป็นขั้นตอน สามารถแก้ไขได้ตรงจุด

•เป็น Open source ของดีแถมฟรีด้วย

•มี Library ให้นำไปใช้เป็นจำนวนมาก

•ง่ายต่อการนำไปขยายและพัฒนาต่อ (extend)

**ข้อเสียของ Rails**

•การติดตั้ง Rails บนเครื่อง development ยุ่งยากกว่าเครื่องมืออื่นๆ เช่น PHP, Java หรือ .NET

•การทำงานช้ากว่าคู่แข่งอยู่มาก พยายามแก้ปัญหานี้จากหลายๆ ฝ่าย เช่น JRuby เป็นการใช้ Java Virtual Machine มารันโปรแกรมที่เขียนขึ้นด้วยภาษา Ruby อาศัยการทำงานของ Java Hotspot Compiler ในการช่วยเพิ่มความเร็วในการทำงาน

•Rails มีทัศนคติที่ไม่ดีกับ Microsoft Windows การส่งเสริมการ ทำงานบน Windows จึงมีไม่มากนัก ทำให้การพัฒนา Ruby on Rails ต้องทำบน Linux หรือ Apple Mac OS เป็นหลักและ Deploy ขึ้น Linux Server เป็นหลัก

9. Heroku เป็น Platform as a Service (Paas) ที่ให้เราใช้งานได้ฟรี (มีแบบเสียเงินด้วย) โดยรองรับภาษาโปรแกรมที่หลากหลาย เช่น Ruby, PHP, Node.js, Python, Java, Clojure, Scala และยังสามารถสร้าง buildpack สำหรับภาษาอื่นๆ เช่น Lua ที่รันอยู่บน OpenResty ได้อีกด้วย

10.เพื่อให้นิสิตได้ศึกษาเกี่ยวกับการพัฒนา software อาจารย์ได้สอนในรูปแบบของภาษา ruby และได้มีการฝึกใช้ Github ซึ่งทำให้นิสิตได้เรียนรู้และได้ลงมือปฎิบัติทำให้เกิดความเข้าใจสามารถนำไปต่อยอดได้