1. อาจกล่าวได้ว่า Agile Method ไม่มีอะไรที่เหมือนกับ Waterfall Model เลย แต่ Waterfall Model ก็ยังเป็นวิธีการพัฒนาระบบที่ใช้กันอยู่ค่อนข้างแพร่หลาย ซึ่งวิธีนี้จะมีลำดับขั้นตอนที่ตายตัว เริ่มตั้งแต่รวบรวมข้อมูล กำหนดความต้องการของผู้ใช้ วิเคราะห์ทางเลือก ออกแบบ เขียนโปรแกรม ทดสอบระบบ และสุดท้ายทำการติดตั้งระบบ โดยแต่ละส่วนของขั้นตอนดังกล่าวจะถือเป็นตัววัดความก้าวหน้าของงาน ปัญหาสำคัญของ Waterfall Model คือขั้นตอนของการพัฒนาที่ไม่ยืดหยุ่น เพราะตัวงานจะแบ่งเป็นช่วงๆแบบตายตัว ทำให้มีข้อผูกมัดตั้งแต่เริ่มโครงงานและไม่สามารปรับเปลี่ยนความต้องการผู้ ใช้ได้ หมายความว่าการพัฒนาโดยใช้ Waterfall Model นั้น ไม่เหมาะกับงานที่ความต้องการของผู้ใช้เข้าใจยาก และมีแนวโน้มว่าจะเปลี่ยนแปลงตลอดเวลา ในทางกลับกัน Agile Method จะแบ่งงานออกเป็นส่วนย่อยๆ แล้วค่อยๆ ทดสอบไปเรื่อยๆทุกสัปดาห์หรือทุกสองสัปดาห์ ทั้งนี้จะเน้นสร้างส่วนย่อยที่สุดของงานทั้งหมดที่สามารถใช้งานได้ทีละชิ้น เพื่อให้ส่งมอบได้รวดเร็วและทำการปรับปรุงเพิ่มเติมอย่างต่อเนื่องตลอดช่วง เวลาของโครงงาน

2. คือ Git เป็น distributed version control system ที่เราสามารถแยก version control ออกเป็น repo ย่อยๆ ได้ แต่ละ repo ย่อยก็สามารถทำ versioning ของตัวเองได้อีกที (เป็น local repo ที่ไม่เกี่ยวกับ remote repo) ส่วน SVN นั้นเป็น centralized version control system คือ version ทั้งหมดจะถูกเก็บที่ remote repo ดังนั้น version ของทุกคนในทีมก็จะตรงกันทั้งหมด ไม่มี local repo

ประเด็นนี้ก็เลยทำให้ Git สามารถทำงานแบบ offline ได้ คือ clone repo มาแล้วจะแก้ไขโค้ดแล้ว commit ไปยัง local repo ก่อนได้ ส่วน SVN นั้น ทุกครั้งที่ commit นั้นต้องไปอัพเดท version กับ server (commit ไปยัง SVN server) ถ้า offline อยู่ หรือเชื่อมต่อกับ server ไม่ได้ ก็จะไม่สามารถ commit ได้

3. rails new feature 1

Git commit –m

Git remote add origin

Git push –u origin master

4. merge conflict คือ เกิดกรณีที่มีการแก้ไขไฟล์เดียวกัน เวลาเดียวกัน Git จะไม่รู้ว่าเราควรจะใช้ไฟล์ไหนดี

ในการพัฒนา software ส่วนใหญ่ต้องทำงานเป็นทีมนั่นคือต้องมีคนทำงานร่วมกันมากกว่า 1 คน ต้องทำงานบน source code ชุดเดียวกัน ดังนั้นมีบ่อยครั้งที่ทำการแก้ไข source code เดียวกัน พร้อม ๆ กัน ส่งผลให้เกิดปัญหา คือ Merge conflict

5. abcde

6. เพราะ Web Application ข้อเสียที่เห็นได้ก็มีดังเช่น ไม่เหมาะสมสำหรับโปรแกรมที่ออกแบบมาเพื่อใช้งานกับข้อมูลส่วนบุคคลที่ไม่จำเป็นต้องแบ่งปันให้กับผู้อื่น รวมถึงข้อมูลที่อาจจะมีความลับสูง (ถ้าต้องส่งผ่านอินเทอร์เน็ต ที่แม้จะเข้ารหัสไว้แล้ว แต่อาจจะถูกเจาะและถอดรหัสนำข้อมูลออกมาไปใช้ได้)

7. เมื่อมีการแบ่งหน้าที่ ความรับผิดชอบกันอย่างชัดเจน เช่นนี้ การจัดการข้อมูล กลไกการทำงาน และการแสดงผล ก็สามารถสลับปรับเปลี่ยนได้โดยไม่ต้องกังวลว่าจะส่งผลกระทบกับการทำงาน ส่วนอื่นๆ

1.เบราว์เซอร์ที่ออกการร้องขอสำหรับ URL /users

2.เส้นทาง /users ไปยังดัชนีการกระทำใน controller

3.การดำเนินการแบบผู้ใช้ในการดึงข้อมูล (User.all)

4.แบบจำลองผู้ใช้ดึงจากฐานข้อมูลผู้ใช้

5.รูปแบบผู้ใช้ส่งกลับรายการของผู้ใช้ควบคุม

6.ตัวควบคุมจับผู้ใช้ในตัวแปร @users ซึ่งจะถูกส่งผ่านไปยัง index

7.แสดงหน้า Ruby เป็น HTML

8.ตัวควบคุมผ่าน HTML กลับไป browser

…English…

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. Web Design Frameworkคือ ชุดโค้ดที่ช่วยให้นักพัฒนาส่วน Front-End ทำงานได้สะดวกขึ้นครับ โดยเค้าจะรวบรวมโค้ด HTML / CSS สำเร็จรูปในการวาง Layout, ตัวอักษร, ปุ่ม, เมนูบาร์มาให้เรียบร้อย โดยหน้าที่ของเราคือแค่ก็อปโค้ดส่วนที่เราต้องการไปใช้ได้เลยข้อเสียคือการใช้ framework มีข้อเสียคือระยะเวลาการศึกษา (learning curve) สูง ดังนั้นอาจจะเป็นการเสียเวลาเกินจำเป็นถ้าเรานำมาใช้กับงานที่ไม่เหมาะสมต่างจาก Rails มีการใช้งานที่ง่ายและรวดเร็ว ลดปัญหางานทีต้องทำซ้ำๆ ทำให้ได้ productivity ที่สูงขึ้น สอดคล้องกับ Methodology

Ruby on Rails จัดเป็น Web Application Framework กล่าวคือเป็นโครงสร้างสำหรับพัฒนา Web Application ประกอบด้วยชุดเครื่องมือต่างๆ สำหรับอำนวยความสะดวกให้แก่วิศกรซอฟต์แวร์ ข้อดีของการใช้ Ruby on Rails มีหลายประการคือ

- ไม่ต้องเสียเวลาพัฒนาเครื่องมือเพื่อใช้เอง จึงสามารถใช้เวลาไปกับการสร้างซอฟต์แวร์ตาม ข้อกำหนดได้มากขึ้น

- ส่งเสริมให้วิศวกรซอฟต์แวร์มีระเบียบวินัยที่ชัดเจน เป็นมาตรฐานเดียวกัน

- มีระบบการทดสอบด้วยซอฟต์แวร์ ทำให้การทดสอบครบถ้วนสม่ำเสมอ

- Ruby on Rails เป็นซอฟต์แวร์ open source ทำให้สามารถตรวจสอบซอร์สโค้ดได้ครบ ถ้วน ช่วยให้การแก้ปัญหาเป็นไปได้ง่ายขึ้น

- ข้อดีอีกข้อของ open source software คือการมีเครือข่ายวิศวกรซอฟต์แวร์ทั่วโลก ช่วยกันประดิษฐ์เครื่องมือเพื่อการใช้งานจำเพาะต่างๆ และแบ่งปันให้เพื่อนวิศวกรซอฟต์แวร์ ได้นำไปใช้และพัฒนาต่อยอดยิ่งๆ ขึ้นไป

9. Heroku คือผู้ให้บริการ Platform as a Services (PaaS) เช่น แอพพลิเคชันเซิร์ฟเวอร์, ดาต้าเบสเซิร์ฟเวอร์ หรือมิดเดิลแวร์อื่นๆ ไม่ต้องเสียเวลาหา software ไม่ต้องหา server และลดความยุ่งยากในการ configuration เพราะเพียงแค่คลิกเลือกภาษาที่ต้องการสร้าง app ไม่ถึงนาทีเราก็มี environment พร้อมใช้งาน ในตอนนี้ใครทำแอป หรือเว็บแอปแล้ววางเครื่องเซิร์ฟเวอร์เองบ้าง ก็อาจจะยังมีอยู่ แต่หลายๆคน อยากลดเวลาในการพัฒนาแอปโดยตั้งเป้าหมายไปที่การพัฒนาแอปเลย ไม่ต้องมาตั้งค่าเครื่องเซิร์ฟเวอร์เองให้เปลืองเวลา เพราะอาจจะไม่มีทรัพยากรณ์มากพอ ทั้งในด้านบุคคลและอื่นๆ อยากจะมุ่งเน้นไปที่พัฒนาโปรแกรมเพียงอย่างเดียว Heroku เป็น Platform as a Service (Paas) ที่ให้เราใช้งานได้ฟรี (มีแบบเสียเงินด้วย) โดยรองรับภาษาโปรแกรมที่หลากหลาย เช่น Ruby, PHP, Node.js, Python, Java, Clojure, Scala และยังสามารถสร้าง buildpack สำหรับภาษาอื่นๆได้ เช่น Lua ที่รันอยู่บน OpenResty ได้อีกด้วย

10. วิชานี้เน้นเรื่องพัฒนาระบบในรูปแบบของภาษาRubyและการใช้Githubซึ่งเป็นเรื่องดีและทันสมัยที่จะทำให้นิสิตได้เรียนรู้กับโปรแกรมใหม่ๆและได้ลงมือปฎิบัติจริงๆ