พื้นฐานของการพัฒนาซอฟต์แวร์และกระบวนการ

1. Agile เป็นการออกแบบจากทีมงานพัฒนาที่มีการบริหารกันเองพร้อมรับความต้องการของลูกค้าที่เปลี่ยนแปลงอยู่ตลอดเวลา ส่งมอบงานที่ใช้งานได้อย่างต่อเนื่องตั้งแต่ทุกๆสัปดาห์ ไปจนถึงทุกๆเดือน โดยมีชิ้นงาน (ซอฟท์แวร์) ที่ใช้งานได้เป็นตัววัดตัวแรกของความก้าวหน้า บุคลากรที่เป็นผู้ใช้และบุคลากรที่พัฒนาระบบจะต้องทำงานร่วมกันอย่างใกล้ชิดในการ พูด คุย พบปะ เน้นการพัฒนาแบบค่อยเป็นค่อยไป มีความยืดหยุ่นต่อการเปลี่ยนแปลง

ส่วน Waterfall ก็จะมีลำดับขั้นตอนที่ตายตัว เริ่มตั้งแต่รวบรวมข้อมูล กำหนดความต้องการของผู้ใช้ วิเคราะห์ทางเลือก ออกแบบ เขียนโปรแกรม ทดสอบระบบ และสุดท้ายทำการติดตั้งระบบ โดยแต่ละส่วนของขั้นตอนดังกล่าวจะถือเป็นตัววัดความก้าวหน้าของงาน

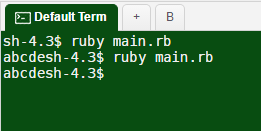
1. VCS หรือ SVN คือการเก็บไฟล์ที่เวลาเราเขียนโค้ดหรือต้องการเก็บไฟล์​ บางทีเราอาจต้องการติดตามการเปลี่ยนแปลง เราอยากกลับไปดูได้ว่ามีอะไรเปลี่ยนแปลงไปบ้างในแต่ละเวอร์ชันของไฟล์ โดยวิธีการจัดเก็บไฟล์โดยใช้ชื่อประมาณแบบนี้ my\_document\_v3\_update.xls แต่ถ้าใช้ Git ปัญหาการใช้เซฟไฟล์แบบใส่เวอร์ชันท้ายไฟล์ก็จะหายไปโดยสิ้นเชิง เพราะเราสามารถย้ายกลับไปยังเวอร์ชันต่างๆ ของไฟล์ได้ คือเราเก็บแค่ my\_document.xls และก็กลับไปดู revision ต่างๆ ที่เราเคยแก้ไว้ได้
2. rails new feature 1

Git commit –m

Git remote add origin

Git push –u origin master

1. ในการพัฒนา software ส่วนใหญ่ต้องทำงานเป็นทีมนั่นคือต้องมีคนทำงานร่วมกันมากกว่า 1 คนดังนั้นมีบ่อยครั้งที่ทำการแก้ไข source code เดียวกัน พร้อม ๆ กันส่งผลให้เกิดปัญหา คือ Merge conflict นั้นเอง บางครั้งก็แก้ไขผิดๆ ถูกๆ บางครั้งทำให้ bug ที่แก้ไขไปแล้ว กลับมาอีกบางครั้งทำให้ feature ที่เคยทำงานได้ ทำงานไม่ได้ เกิดจากการที่ไม่พูดคุยตกลงกันให้ดีเสียก่อนว่าใครจะ ทำงานส่วนไหน



1. Web Application นั้นมีการใช้งานที่ง่าย แต่มี feature การใช้งานที่ค่อนข้างจำกัด ต่างจากการใช้ซอฟต์แวร์ในเครื่องคอมพิวเตอร์ที่มี feature หลากหลายมากกว่าตรงกับความต้องการใช้มากกว่า Web Application
2. เมื่อมีการแบ่งหน้าที่ ความรับผิดชอบกันอย่างชัดเจน การจัดการข้อมูล กลไกการทำงาน และการแสดงผล ก็สามารถสลับปรับเปลี่ยนได้โดยไม่ต้องกังวลว่าจะส่งผลกระทบกับการทำงาน ส่วนอื่นๆ ตัวอย่าง

* การแสดงผลของข้อมูลเป็น HTML หรือ XML สามารถทำได้ด้วย View ที่แตกต่างกันสองชุด แต่สามารถแสดงข้อมูลชุดเดียวกันได้โดยไม่ต้องเปลี่ยน Model
* การสลับ Database Back end จาก MySQL เป็น MongoDB ได้โดยการทำงานของระบบ ทั้งหมดยังคงเดิม
* การเปลี่ยนขั้นตอนการสมัครสมาชิกโดยที่การเก็บข้อมูลยังเหมือนเดิม และการแสดงผลลัพธ์ ก็ไม่เปลี่ยนแปลง

การแยกส่วนการทำงานเช่นนี้ทำให้การทดสอบระบบเป็นไปได้ง่ายขึ้น โดยเริ่มจาก การทำ Unit Test คือการทดสอบแยกส่วน ภายใน แต่ละส่วน, จากนั้นทำ Functional Test คือการ ทดสอบการทำงานร่วมกันของส่วนประกอบภายในทั้งหมด, แล้วจึงทดสอบ Integration Test คือ การทดสอบการทำงานร่วมกันของ Application ที่เกี่ยวข้องต่างๆ

1. Rails เป็น Web Framework มีลักษณะเป็น MVC (Model-View-Controller) ถูกออกแบบมาให้มีการใช้งานที่ง่ายและรวดเร็ว ลดปัญหางานทีี่ต้องทำซ้ำๆ ทำให้ได้ productivityที่สูงขึ้น สอดคล้องกับ Methodology แบบ Agile และมีแนวคิดพื้นฐานคือ

* Don't Repeat Yourself (DRY) อะไรที่อยู่แล้ว ก็ไม่มีความจำเป็นต้องทำสิ่งนั้นซ้ำ

Reuseable นั้นเอง

* Convention over configuration คือข้อตกลงที่ Rails สร้างขึ้นเป็นแบบแผน ที่กำหนดมา

ขอเพียงเราทำตามรูปแบบเหล่านั้นผลลัพธ์จะออกมาตามที่เราต้องการ

1. Heroku เป็น Platform as a Service (Paas) โดยรองรับภาษาโปรแกรมที่หลากหลาย เช่น

* RubY
* PHP
* Node.js
* Python
* Java
* Clojure
* Scala
* สามารถสร้าง buildpack สำหรับภาษาอื่นๆได้

นอกจากรองรับภาษาโปรแกรมที่หลากหลายแล้ว ตัว Heroku มี App Store ของมันด้วยเรียกว่า add-ons สำหรับเพิ่มเติมบริการอื่นๆเข้าไปในแอปของเรา เช่น PostgreSQL, MongoDB, Redis เป็นต้น ซึ่งก็มีทั้งฟรี และไม่ฟรีให้เลือกใช้งาน

1. เพื่อเป็นแนวคิดที่สำคัญในการเรียนรู้และเพื่อให้นิสิตได้รู้จักการใช้งานของซอฟ์แวร์แต่ละชนิด ได้มองเห็นถึงจุดเด่นและประโยชน์ในการใช้ซอฟท์แวร์ที่หลากหลายทั้ง StoriesOnBoard, Gitและ Github , Ruby , Rails framework และ Heroku