**1.** 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.**ไม่เห็นด้วยเพราะมีความเป็นไปได้ที่ cvs จะไม่นำมาใช้แล้วแต่คงไม่ถึงขนาดที่จะต้องสูญพันธุ์เพราะ svn ยังคงสามรถใช้งานได้อยู่เพราะเป็น version control เหมือนกัน

**4**.เมื่อไร merge : เมื่องานที่ branch เสด ฟิวเจอร์เสด

เมื่อไรมี conflict : เมือ่โปรแกรมเมอร์สองคนทำงานในส่วนเดี๋ยวกัน ต้องการ merge พร้อมกัน

แก้ conflict อย่างไร : คุยกันว่าจะเอาโค้ดเเบบไหนดี , หรือให้โปรเจคเมเนเจอร์ตัดสินเลยว่าจะเอาโค้ดไหน

การป้องกันconflict : การทำ sw engineering ที่ดีโดยมีการแบ่งงานกันที่ชัดเจน

6.ส่วนตัวแล้วการทำweb Application เราสามารถทำงานกับมันได้ทุกที่และยังสามารถแบ่งปันให้กับคนอื่นได้อีกด้วยมันไม่ใช่การตามแฟชั่น

7. Model ส่วนของข้อมูลทั้งหมด

View ส่วนของหน้าจอที่เเสดงให้ผู้ใช้เห็น

Controller ตัวคอยควบคุมระหว่าง m v เพื่อให้สื่อสารกันผ่าน control

1.Client หรือผู้ใช้ร้องขอข้อมูลทาง Web site

2.Web server รับการร้องขอในรูปแบบต่างๆ ผ่านทาง HTTP, RSS, ATOM หรือ SOAP

3.Web server ส่งต่อไปยังไฟล์ที่ชื่อว่า dispatcher ที่อยู่ใน Rails ซึ่งเป็นไฟล์สำหรับโหลด Controller ให้ทำงาน

4.ไฟล์ Dispatcher ทำงาน โดยจะโหลดตัว Controller ที่อยู่ใน Rails ขึ้นมา

5.Controller เป็นส่วนแรกของการทำงานแบบ MVC ติดต่อกับ Model และ View หน้าที่คือการประมวลผลหลักของ

8**.Bootstrap Framework**

**ข้อดี**

* สามารถศึกษาได้ด้วยตนเอง
* ไม่ต้องเขียนโค้ดเอง
* ไม่ต้องมีการติดตั้ง

**ข้อเสีย** ถ้ามีคนมาทำงานต่ออาจจะยุ่งยาก

**Rails Framework**

**ข้อดี**

* ไม่ต้องเขียนโค้ดเองทำให้คนอื่นทำงานต่อได้ง่าย

**ข้อเสีย**

* การติดตั้ง Ruby on Rails บนเครื่อง development ยุ่งยากกว่าเครื่องมืออื่นๆ เช่น PHP, Java หรือ .NE
* การ Deploy Rails application ขึ้นเซิร์ฟเวอร์จริง ก็ยิ่งยุ่งยากมากกว่าการติดตั้งบนเครื่อง development
* Ruby on Rails ได้รับความนิยมมากในต่างประเทศ แต่ในประเทศไทยเพิ่งเริ่มมีความนิยมในระยะเวลา 1-2 ปีมานี้ ทำให้ตำรับตำราต่างๆ เป็นภาษาต่างประเทศทั้งหมด

**9**. Heroku เป็น Platform as a Service (Paas) ที่ให้เราใช้งานได้ฟรี (มีแบบเสียเงินด้วย) โดยรองรับภาษาโปรแกรมที่หลากหลาย เช่น Ruby, PHP, Node.js, Python, Java, Clojure, Scala และยังสามารถสร้าง buildpack สำหรับภาษาอื่นๆได้ ไม่ต้องเสียเวลาหา software ไม่ต้องหา server และลดความยุ่งยากในการ configuration เพราะเพียงแค่คลิกเลือกภาษาที่ต้องการสร้าง app ไม่ถึงนาทีเราก็มี environment พร้อมใช้งาน ที่สำคัญฟรี Heroku จะทำผ่าน heroku toolbelt ส่วนของโค้ดที่จะเขียน ก็ใช้ text editor ได้ตามความถนัด

10.เพราะในสายการทำงานหรือการเรียนเราต้องสามารถพัฒนาหรือสร้างApplicationสำหรับการใช้งานได้จำเป็นต้องมีการเรียนรู้เกี่ยวกับ Software Development