1. Agile เป็นหลักการในการพัฒนา software แบบใหม่ที่เน้นความสามารถในการปรับเปลี่ยนได้อย่างรวดเร็วตามความเป็นจริงที่เปลี่ยนแปลงอยู่เสมอ มีความยืดหยุ่นในการเปลี่ยนแปลง เมื่อความต้องการของผู้ใช้เปลี่ยนแปลงไป ทีมงานพัฒนาก็ต้องพร้อมที่จะรับการเปลี่ยนแปลงตามไปด้วย อาจกล่าวได้ว่า Agile ไม่มีอะไรที่เหมือนกับ Waterfall เลย แต่ Waterfall ก็ยังเป็นวิธีการพัฒนาระบบที่ใช้กันอยู่ค่อนข้างแพร่หลาย ซึ่งวิธีนี้จะมีลำดับขั้นตอนที่ตายตัว เริ่มตั้งแต่รวบรวมข้อมูล กำหนดความต้องการของผู้ใช้ วิเคราะห์ทางเลือก ออกแบบ เขียนโปรแกรม ทดสอบระบบ และสุดท้ายทำการติดตั้งระบบ โดยแต่ละส่วนของขั้นตอนดังกล่าวจะถือเป็นตัววัดความก้าวหน้าของงาน ปัญหาสำคัญของ Waterfall คือขั้นตอนของการพัฒนาที่ไม่ยืดหยุ่น เพราะตัวงานจะแบ่งเป็นช่วงๆแบบตายตัว ทำให้มีข้อผูกมัดตั้งแต่เริ่มโครงงานและไม่สามารถปรับเปลี่ยนความต้องการผู้ ใช้ได้ หมายความว่าการพัฒนาโดยใช้ Waterfall นั้น ไม่เหมาะกับงานที่ความต้องการของผู้ใช้เข้าใจยาก และมีแนวโน้มว่าจะเปลี่ยนแปลงตลอดเวลา ถ้าค้นพบข้อผิดพลาดของขั้นที่เสร็จสิ้นแล้ว ไม่สามารถแก้ไขได้ การแก้ไขจำเป็นต้องเริ่มรอบ (Iteration) ใหม่ ระหว่างการทำCoding เจอข้อผิดพลาดในงานออกแบบในความเป็นจริง หลังการทำงานในแต่ละขั้นควรสามารถย้อนไปแก้ไขความผิดพลาดในขั้นใดใดก็ได้ก่อนหน้า ในทางกลับกัน Agile จะแบ่งงานออกเป็นส่วนย่อยๆ แล้วค่อยๆ ทดสอบไปเรื่อยๆทุกสัปดาห์หรือทุกสองสัปดาห์ ทั้งนี้จะเน้นสร้างส่วนย่อยที่สุดของงานทั้งหมดที่สามารถใช้งานได้ทีละชิ้น เพื่อให้ส่งมอบได้รวดเร็วและทำการปรับปรุงเพิ่มเติมอย่างต่อเนื่องตลอดช่วงเวลาของโครงงาน
2. Git และ Githubในเรื่องการทำ Version Control เป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็คเรา มีการ backup code ให้เราสามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของโปรเจ็คที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ ฉะนั้น Version Control ก็เหมาะอย่างยิ่งสำหรับนักพัฒนาไม่ว่าจะเป็นคนเดียวโดยเฉพาะอย่างยิ่งจะมีประสิทธิภาพมากหากเป็นการพัฒนาเป็นทีม
3. git add <ไฟล์ที่ต้องการส่ง> git commit –m “ข้อความที่ต้องการส่ง” git remote add origin [https://github.com/ชื่อ](https://github.com/%E0%B8%8A%E0%B8%B7%E0%B9%88%E0%B8%AD) github ID ของเรา/final\_887342.git git push –u origin master
4. การ merge และแก้ไขเป็น conflict เมื่อไร merge : เมื่องานที่ branch เสร็จ future เสร็จ เมื่อไรมี conflict : เมื่อโปรแกรมเมอร์สองคนทำงานในส่วนเดียวกัน ต้องการ merge พร้อมกัน แก้ conflict อย่างไร : คุยกันว่าจะเอาโค้ดเเบบไหนดี , หรือให้โปรเจคเมเนเจอร์ตัดสินเลยว่าจะเอาโค้ดไหน การป้องกัน conflict : การทำ sw engineering ที่ดีโดยมีการแบ่งงานกันที่ชัดเจน โดยปกติแล้ว git merge จะรวมโค้ดให้เราเองอัตโนมัติ แต่ก็จะมีข้อยกเว้นเมื่อ แก้ไขไฟล์เดียวๆกัน ลองนึกถึงกรณีที่เราและเพื่อนร่วมทีม แก้ไขไฟล์เดียวกัน Git จะเกิดการ conflict เมื่อเราจะ merge โค้ด โดยไม่รู้ว่าจะใช้โค้ดของเราหรือของเพื่อน วิธีแก้ก็คือ ทำการ edit แล้ว commit ไปใหม่
5. ผลลัพธ์ คือ abcde"a".."e"
6. การพัฒนาระบบงานบนเว็บ ซึ่งมีข้อดีคือ ข้อมูลต่าง ๆ ในระบบมีการไหลเวียนในแบบ Online ทั้งแบบ Local (ภายในวงLAN) และ Global(ออกไปยังเครือข่ายอินเตอร์เน็ต) ทำให้เหมาะสำหรับงานที่ต้องการข้อมูลแบบ Real Time ระบบมีประสิทธิภาพแต่ใช้งานง่าย เหมือนกับท่านทำกำลังท่องเว็บ ระบบงานที่พัฒนาขึ้นมาจะตรงกับความต้องการกับหน่วยงาน หรือห้างร้านมากที่สุด ไม่เหมือนกับโปรแกรมสำเร็จรูปทั่วไป ที่มักจะจัดทำระบบในแบบกว้าง ๆ ซึ่งมักจะไม่ตรงกับความต้องการที่แท้จริง ระบบสามารถโต้ตอบกับลูกค้า หรือผู้ใช้บริการแบบ Real Time ทำให้เกิดความประทับใจ เครื่องที่ใช้งานไม่จำเป็นต้องติดตั้งโปรแกรมใด ๆ เพิ่มเติมทั้งสิ้น ตัวอย่างระบบงานที่เหมาะกับเว็บ แอพพลิเคชั่น เช่น ระบบการจองสินค้าหรือบริการต่าง ๆ เช่น การจองที่พัก การจองโปรแกรมทัวร์ การจองแผ่น CD-DVD ฯลฯ ระบบงานบุคลากร ระบบงานแผนการตลาด ระบบการสั่งซื้อแบบพิเศษ ระบบงานในโรงเรียน เช่น ระบบงานวัดและประเมินผล ระบบงานปกครอง ระบบงานห้องสมุด ระบบการลงทะเบียน เช็คเกรด ฯลฯ ระบบงานอื่น ๆ ที่ต้องการนำข้อมูลมา Online
7. model view controller (MVC) Model ส่วนของข้อมูลทั้งหมด View ส่วนของหน้าจอที่เเสดงให้ผู้ใช้เห็น Controller ตัวคอยควบคุมระหว่าง m v เพื่อให้สื่อสารกันผ่าน control
8. ยกตัวอย่าง framework ที่เคยใช้งานคือ framework ระบบสั่งซื้อออนไลน์ Framework เป็นโครงสร้างของโปรแกรม โปรแกรมเมอร์ส่วนใหญ่ ใช้เวลาส่วนมากของการพัฒนา web application ไปกับการเขียนโปรแกรมโดยทั่วไป มากกว่าฟังก์ชันพิเศษ ดังนั้นการนำชิ้นส่วนของ code เดิมกลับมาใช้ (Reusable) จึงได้รับความนิยม ในการสร้าง web application โดยตัว Component ที่นำกลับมาใช้ใหม่นี้ จะเรียกว่า "framework" Framework ที่ดีต้องง่ายในการเพิ่ม แก้ไข เปลี่ยนแปลง ในส่วนของ application Rails framework : เป็น Web Framework มีลักษณะเป็น MVC (Model-View-Controller) Rails ถูกออกแบบมาให้มีการใช้งานที่ง่ายและรวดเร็ว ลดปัญหางานที่ต้องทำซ้ำๆ ทำให้ได้ productivity ที่สูงขึ้น สอดคล้องกับ Methodology แบบ Ajile และมีแนวความคิดพื้นฐานคือ อะไรที่อยู่แล้ว ก็ไม่มีความจำเป็นต้องทำสิ่งนั้นซ้ำ Reuseable นั้นเอง ข้อตกลงที่ Rails สร้างขึ้นเป็นแบบแผน ที่กำหนดมาขอเพียงเราทำตามรูปแบบเหล่านั้นผลลัพธ์จะออกมาตามที่เราต้องการ โดยที่ไม่จำเป็นต้องใช้การ configuration สำหรับคนที่ใช้ Convention ได้อย่างคล่องแคล่วและถูกต้องแล้วจะทำให้การพัฒนาระบบได้เร็วขึ้นเพราะไม่ต้องมาทำการ configuration ให้ยืดยาว แต่มีข้อดีก็ต้องมีข้อเสียครับ เพราะ Convention ถ้าเป็นมือใหม่หัดเขียนคงต้องศึกษาและทำความเข้าใจกันมากซักหน่อย และมันอาจสร้างความประหลาดใจให้คุณได้
9. 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 ได้
10. เป็นวิชาเลือกที่มีความรู้เบื้องต้นเกี่ยวกับวิศวกรรมซอฟต์แวร์ กระบวนการพัฒนาซอฟต์แวร์ การบริหารโครงการซอฟต์แวร์ กระบวนการวิศวกรรมความต้องการ แบบจำลองระบบ การออกแบบ การสร้างซอฟต์แวร์ การทดสอบ การตรวจสอบความถูกต้อง ตัวชี้วัดซอฟต์แวร์ การประกันคุณภาพซอฟต์แวร์ การจัดการและควบคุมการเปลี่ยนแปลงใน การพัฒนางานด้านซอฟต์แวร์ การบำรุงรักษาซอฟต์แวร์ รู้หลักการทำงาน MVC การใช้ภาษา Ruby อย่างง่ายแทน ภาษา Java มีการใช้ gitthub เป็นประโยชน์ในการทำงานโปรเจค สามารถนำมาประยุกต์ใช้ในสาขาวิชาวิทยาการสารสนเทศได้