Final\_part\_1

1. วิจารณ์

* Agile เป็นกระบวนการ software แบบใหม่ที่พัฒนาไว ใช้งานได้รวดเร็ว
* Waterfall เป็นกระบวนการที่ทำเป็นลำดับขั้นตอน ใช้เวลานาน มีความแน่นอนชัดเจน
* ไม่ควรยึดติดกับ software เพียงตัวใดตัวหนึ่ง แต่ควรเลือกใช้ให้เหมาะสมกับการทำงานของแต่ล่ะบุคคลมากกว่า

1. วิจารณ์

* โดยส่วนตัวคิดว่าการใช้ GitHub มีความสะดวกสบายกว่าการที่ต้องมาแยกเก็บโค้ดใน CVS หรือ SVN เพราะการใช้ GitHub สามารถเก็บไฟล์ได้หลายอย่างไม่ใช่แค่โค้ดเพียงอย่างดียว

1. คำสั่ง

* rails new frature1
* git commit -m "Initialize repository"
* git remote add origin <https://github.com/your_github_id/feature1.git>
* git push –u origin master

1. วิจารณ์

* การ merge ไม่จำเป็นต้องเกิด conflict ทุกครั้งก็ได้ แต่ไม่แปลกที่จะเกิดการ merge conflict เพราะการพัฒนา software ส่วนใหญ่ต้องทำงานกันเป็นทีมจึงมีปัญหาที่จะใช้ source code เดียวกันพร้อมๆกัน

1. ผลลัพธ์ที่ได้

* abcde”a”..”e”

1. วิจารณ์

* โดยส่วนตัวคิดว่าไม่ใช่เรื่องที่น่าประหลาดใจเท่าไหร่ และไม่ได้ตามแฟชั่น เพราะสมัยนี้มีการพัฒนา software ต่างๆมากมายมาให้เราสามารถเลือกใช้ได้ตามความต้องการ แต่เราก็สามารถพัฒนา software ของตัวเองชึ้นมาใช้ตามต้องการได้

1. การทำงานของ MVC
2. เริ่มจาก User ส่ง Request ไปที่ Web App ซึ่งถูกส่งต่อให้ Controller
3. แล้ว Controller จะเรียก Method ให้ทำงานเพื่อจัดการ Request นั้น
4. Model จะทำการคำนวณและอาจติดต่อกับ Database เพื่อจัดการกับ Request    
   นั้น แล้วส่งผลลัพธ์กลับไปที่ Controller
5. เมื่อ Controller ได้ผลลัพธ์จาก Model แล้วก็ใช้ผลลัพธ์นั้นส่งต่อให้ View ทำงาน
6. View จะสร้าง Page สำหรับแสดงผลลัพธ์นั้น แล้วส่ง page กลับไปที่ Controller
7. Controller ส่ง Page นั้น (เป็น Response) กลับไปยัง User
8. Framework : Web Application Framework

* Rails เป็น Framework อาจมองว่ามีการวางกรอบพัฒนามาอย่างคร่าวๆ เอาไว้ให้การพัฒนา software ขึ้นมามีระเบียบที่ดีขึ้น

ข้อดี

* ถูกออกแบบมาให้มีการใช้งานง่ายและรวดเร็ว
* อะไรที่มีอยู่แล้วก็ไม่จำเป็นต้องทำสิ่งนั้นซ้ำ

ข้อเสีย

* ถ้าเป็นมือใหม่หัดเขียนต้องศึกษาและทำความเข้าใจกันมากหน่อย

1. Heroku คือ Platform as a Service (Paas) ที่ให้เราใช้งานได้ฟรี โดยรองรับโปรแกรมที่หลากหลาย

* มีบทบาทอย่างมากในการพัฒนา Web software ของเรา

1. เพื่อให้นักศึกษาได้เรียนรู้ความรู้เบื้องต้นเกี่ยวกับวิศวกรรมซอฟต์แวร์ กระบวนการพัฒนาซอฟต์แวร์ การบริหารโครงการซอฟต์แวร์ กระบวนการวิศวกรรมความต้องการ แบบจำลองระบบ การออกแบบ การสร้างซอฟต์แวร์ การทดสอบ การตรวจสอบความถูกต้อง ตัวชี้วัดซอฟต์แวร์ การประกันคุณภาพซอฟต์แวร์ การจัดการและควบคุมการเปลี่ยนแปลงใน การพัฒนางานด้านซอฟต์แวร์ การบำรุงรักษาซอฟต์แวร์