1.เเสดงว่ายังไม่เข้าใจ agile ดี agile ไม่ได้เป็นแฮกเกอร์อย่างที่คิด เพราะ agile นั้นเน้นว่าใครถนัดอะไร และการพูดคุยสื่อสารกัน มากกว่า การยึดติดที่เครื่องมือและกระบวนการ เช่นเปลี่ยนให้โปรแกรมเมอร์ไปคุยกับลูกค้าแทน ลูกค้าบอกอะไรมาก็ทำตามนั้นได้เลย

2.เป็นไปไม่ได้อย่างเเน่นอน เพราะ git นั้นมองข้อมูลต่าง ๆ โดยทั่วไประบบอื่นมักจะเก็บข้อมูลในรูปแบบของการแก้ไขที่เกิดขึ้นกับไฟล์ต่าง ๆ CVS จะมองข้อมูลในรูปแบบของไฟล์และการแก้ไขต่าง ๆ ที่เกิดขึ้นกับไฟล์แต่ละไฟล์ ซึ้งถึงเเม้ว่า git จะทรงพลังกว่าเเต่ก้ไม่ได้หมายความว่า CVS จะต้องสูญพันธ์ไป

3. git checkout

git add feature1

git commit -m “create feature1”

git push origin master

4.เมื่องานที่ branch เสร็จเเล้ว ทำงานฟีเจอร์ ต่างๆเสร็จ ทำการ merge ได้เลย เเต่เมื่อต้องทำงานพร้อมกับเพื่อนร่วมงานในส่วนเดียวกันให้conflict

5. ch

6.ไม่ได้เรียกว่าตามแฟชั่น เเต่เป็นการทำ app เเล้วสะดวกต่อการอัพเดตมากกว่า เพราะสามารถที่จะอัพเดต version บน web ได้เลย

7.Model ส่วนของข้อมูลทั้งหมด

View ส่วนของหน้าจอที่เเสดงให้ผู้ใช้เห็น

Contorller ตัวคอยควบคุมระหว่าง Model View เพื่อให้สื่อสารกันผ่าน control

8. php framwork

ข้อดี

1.มีธรรมเนียมการเขียนที่ชัดเจน โปรแกรมเมอร์คนเก่าลาออก คนใหม่เข้ามารู้ Framework ตัวนั้นเขียนต่อได้เลย

2.ลดเวลาในการสร้าง Create, Read, Update และ Delete หรือ CRUD เพราะ Framework ส่วนใหญ่มีระบบ Generator ให้ พูดง่ายๆ ว่า Gen Code Insert Update Delete ให้ได้เลย ลดเวลาในการเขียนส่วนนี้ไปอีก

3.มีระบบในการเชื่อมต่อไปยังฐานข้อมูลได้มากกว่า 1 ชนิด เช่น เชื่อมต่อไปยัง MySQL, MS SQL, ProgreSQL, SQLite และอื่นๆ ได้หลากหลาย (ทั้งนี้ขึ้นกับความสามารถของแต่ละ Framework)

ข้อเสีย

1.มี Class จำนวนมากมาย เพื่อให้การทำงานอย่างเป็นระบบ ทำให้ต้องเขียน Class ที่เป็นตัวช่วยให้ผู้พัฒนาใช้ แต่บาง Class อาจไม่จำเป็น

2.ใช้เวลาในการเรียนรู้และศึกษาธรรมเนียมของการเขียนโปรแกรมใน Framework นั้นๆ (แต่ระยะหลังนี้มีการใช้มาตรฐาน PSR ในการพัฒนา ต่อไปเรียนรู้ Framework ตัวเดียวแต่สามารถเขียนได้ทุก Framework)

Rails feamwork

ข้อดี

1. ไม่ต้องเสียเวลาพัฒนาเครื่องมือเพื่อใช้เอง จึงสามารถใช้เวลาไปกับการสร้างซอฟต์แวร์ตาม ข้อกำหนดได้มากขึ้น
2. ส่งเสริมให้วิศวกรซอฟต์แวร์มีระเบียบวินัยที่ชัดเจน เป็นมาตรฐานเดียวกัน
3. มีระบบการทดสอบด้วยซอฟต์แวร์ ทำให้การทดสอบครบถ้วนสม่ำเสมอ
4. Ruby on Rails เป็นซอฟต์แวร์ open source ทำให้สามารถตรวจสอบซอร์สโค้ดได้ครบ ถ้วน ช่วยให้การแก้ปัญหาเป็นไปได้ง่ายขึ้น
5. ข้อดีอีกข้อของ open source software คือการมีเครือข่ายวิศวกรซอฟต์แวร์ทั่วโลก ช่วยกันประดิษฐ์เครื่องมือเพื่อการใช้งานจำเพาะต่างๆ และแบ่งปันให้เพื่อนวิศวกรซอฟต์แวร์ ได้นำไปใช้และพัฒนาต่อยอดยิ่งๆ ขึ้นไป

ข้อเสีย

1.การติดตั้ง Ruby on Rails บนเครื่อง development ยุ่งยากกว่าเครื่องมืออื่นๆ เช่น PHP, Java หรือ .NET

2.การ Deploy Rails application ขึ้นเซิร์ฟเวอร์จริง ก็ยิ่งยุ่งยากมากกว่าการติดตั้งบนเครื่อง development

3.ความเร็วในการทำงานของภาษา Ruby จัดได้ว่าช้ากว่าคู่แข่งอยู่มาก มีความพยายาม แก้ปัญหานี้จากหลายๆ ฝ่าย เช่น JRuby เป็นการใช้ Java Virtual Machine มารันโปรแกรมที่เขียนขึ้นด้วยภาษา Ruby อาศัยการทำงานของ Java Hotspot Compiler ในการช่วยเพิ่มความเร็วในการทำงาน

4.ทั้ง Ruby และ Ruby on Rails มีทัศนคติที่ไม่ดีกับ Microsoft Windows การส่งเสริมการ ทำงานบน Windows จึงมีไม่มากนัก ทำให้การพัฒนา Ruby on Rails ต้องทำบน Linux หรือ Apple Mac OS เป็นหลักและ Deploy ขึ้น Linux Server เป็นหลัก

5.Ruby on Rails ได้รับความนิยมมากในต่างประเทศ แต่ในประเทศไทยเพิ่งเริ่มมีความนิยมในระยะเวลา 1-2 ปีมานี้ ทำให้ตำรับตำราต่างๆ เป็นภาษาต่างประเทศทั้งหมด

9. Heroku เป็นเหมือน server โดยที่ไม่ต้องติดตั้ง server มีบทบาทโดยการจะ push ไปที่ heroku นั้น ทาง heroku เองจะมี [toolbelt](https://toolbelt.heroku.com/)  เป็น Heroku Client มาให้เราใช้ ซึ่งจะมีทุก platform เลย ไม่ว่าจะเป็น (window ,osx ,linux) แล้วใน toolbelt มันจะมี foreman ติดตั้งมาด้วย ซึ่งมันจะทําหน้าที่เป็น config file ว่าจะให้ Heroku ทํางานยังไง และ มองตัวไหนเป็นตัวที่มันจะเป็นไฟล์หลัก

10.เพราะจะให้นิสิตนักศึกษาที่เรียนอยู่เเล้วจบไปสามารถที่จะพัฒนาศักยภาพตัวเอง เเละสามารถที่จะเก็บผลงานตัวเองที่เคยทำไว้เป็น github ได้ จะมีความน่าเชื่อถือมากกว่าพูดเฉยๆ ว่ามีผลงาน