**นายอนพัทย์ เมืองชล รหัสนิสิต 57160033**

**final\_part\_1**

**ข้อ 1**

**ตอบ** จากคำกล่าวอ้างข้างต้นนี้ สำหรับผมแล้วเห็นว่าเขาเป็นคนที่ค่อนข้างจะปิดใจ ไม่เปิดรับอะไรใหม่ๆที่เข้ามาในชีวิต waterfall เป็นกระบวนการพัฒนาโปรแกรมที่ดี และใช้กันมาอย่างเป็นระบบแบบแผน เข้าใจง่ายไม่ซับซ้อน ก็แค่ทำตามกระบวนการที่วางแผนไว้ไปเรื่อยๆ งานก็จะเสร็จสมบูรณ์ แต่ถ้าเราลองเปิดใจมาดูกระบวนการแบบ agile เราจะพบว่ากระบวนการพัฒนาซอฟแวร์แบบนี้นั้น มีความยืดหยุ่น ประหยัดค่าใช้จ่ายเยอะกว่าหากมีการแก้ไข และตรงตามความต้องการของลูกค้าอย่างแน่นอน จริงอยู่ครับที่ว่า waterfall เป็นกระบวนการพัฒนาโปรแกรมที่ดีอยู่แล้ว แต่คุณจะเลือกอะไรล่ะครับ ระหว่างคำว่า “ดีแล้ว” กับ “ดีกว่า”

**ข้อ 2**

**ตอบ** จากคำกล่าวอ้างข้างต้นนี้ สำหรับผมแล้ว เห็นด้วยครับ สำหรับเรื่องที่ว่า พวก centralized version control แบบ CVS หรือ SVN จะสูญพันธ์ไป เพราะเมื่อเทียบกันแล้วกับ Git และ Github นั้น เราจะรู้ได้เลยว่ามันดีกว่าขนาดไหน แน่นอนแหละครับ ว่าสิ่งที่ยังสู้ไม่ได้ ก็จะสูญหายไป ไม่ต่างกับโทรเลข ที่ต้องยกเลิก เพราะมีอย่างอื่นที่ดีกว่า แต่ก็นั่นแหละครับ ในทางเดียวกันเทคโนโลยี มาไว ไปไวด้วยเช่นกัน ใครจะรู้ครับ ในอนาคตอีก 10 ปีข้างหน้า อาจจะมี version control แบบใหม่ขึ้นมา ที่สามารถทำงานได้ดีกว่า Git และ Github ในปัจจุบันนี้ ก็เป็นได้...

**ข้อ 3**

**ตอบ** git checkout -b feature1

------- เมื่อทำงาน feature1 เสร็จสิ้น -------

git add -A

git commit -m “Success feature1”

git push -u origin feature1

**ข้อ 4**

**ตอบ** จากคำกล่าวอ้างข้างต้นนี้ สำหรับผมแล้ว ไม่เห็นด้วยครับ จริงอยู่ที่การเกิด conflict นั้นเป็นเรื่องที่อาจจะเกิดขึ้นได้เสมอในการพัฒนาโปรแกรมหลายๆคน แต่ปัญหานี้จะเกิดน้อยอย่างแน่นอน ถ้าเรามีการแบ่งหน้าที่ หรือ feature ในการพัฒนาโปรแกรมกันอย่างดี ดังนั้น รุ่นพี่คนนั้น ควรที่จะนำปัญหานี้ไปปรึกษากันกับทีมมากกว่า ไม่ใช่ปล่อยให้เกิดมาตลอด 3 ปี โดยที่ไม่ได้คิดจะแก้ไขอะไรเลย… แต่ส่วนที่ผมเห็นด้วยก็คือ เป็นคำแนะนำที่ดีครับ ทำให้รุ่นน้องนั้น ได้รู้ว่าการมีปัญหาแบบนี้ด้วยในการทำงานจริง เพื่อจะได้ตระหนักถึงมัน ไม่ให้มันเกิดขึ้นอีก นั่นเอง...

**ข้อ 5**

**ตอบ** ผลลัพธ์จากการรันโปรแกรม Ruby ('a'..'e').each {|ch| print ch} นี้คือ

abcde

**ข้อ 6**

**ตอบ** จากคำกล่าวอ้างข้างต้นนี้ สำหรับผมแล้วเห็นว่า เขาคนที่พูดนี้ ไม่ได้เปิดใจรับสิ่งใหม่ๆอีกเช่นเดียวกัน ทั้งสองแบบมีทั้งข้อดี และข้อเสียที่ต่างกันอยู่แล้วครับ เช่น แทบจะเป็นไปไม่ได้ที่ Web Application จะทำงานเร็วว่า ซอฟแวร์ใน CD เช่นเดียวกัน แทบจะเป็นไปไม่ได้ที่ ซอฟแวร์ใน CD ตัวเดียวกัน จะทำงานได้ทุกๆ Platform และสามารถอัปเดทได้รวดเร็ว แบบ Web Application และผมไม่เห็นว่ามันน่าประหลาดใจตรงไหนเลย แต่มันอยู่ที่ว่า งานที่เราทำ หรือซอฟแวร์ที่เราจะพัฒนานั้น มันเหมาะกับการทำงานในรูปแบบไหนมากกว่ากันต่างหาก...

**ข้อ 7**

**ตอบ** จากรูปภาพนี้

เริ่มจาก user ได้เปิดเข้า Web Application Rails จาก web browser จากนั้น web browser จะทำการร้องขอไปยัง server โดยส่งในส่วนของ URI คือ /users จากนั้น Rails router จะทำการร้องขอ Controller ที่ชื่อ users\_controller.rb โดยคำร้องขอคือ index (Rails router นั้นสามารถรู้ได้ว่าต้องร้องขอ Controller ตัวใด เพราะ ผู้ใช้ ร้องขอ URI มากเป็น /users ) หลังจากนั้น users\_controller.rb จะทำการขอข้อมูลจาก Model ที่ชื่อ user.rb โดยมีคำร้องขอคือ User.all เพื่อ Select ข้อมูล Users จาก Database ทั้งหมด แล้วส่งคืนข้อมูลนั้นให้กับ users\_controller.rb จากนั้น users\_controller.rb ก็จะเรียก View ที่ชื่อ index.html.erb โดยเรียกคือ @users เพื่อประมวลผลเราค่าที่ได้จาก Model นั้นใส่ใน View แล้วส่งคืนให้กับ ผู้ใช้ในรูปแบบ HTML นั่นเอง...

**ข้อ 8**

**ตอบ** ถ้าให้ยกตัวอย่าง framework ที่ใช้พัฒนา Web Application ที่กระผมเคยใช้มา และใช้บ่อย นั่นก็คือ Codeigniter สิ่งที่มีเหมือนกันกับ Rails คือการเขียนแบบ MVC แต่สิ่งที่ต่างกัน และผมเห็นว่า Rails นั้นดีกว่า ตรงที่ว่า ใช้งานได้ง่ายมาก!! สามารถทำ Rest API ได้อย่างง่ายดาย ง่ายกว่า Codeigniter เยอะ และยังเป็นการใช้ภาษา Ruby ในการเขียนโค้ด ซึ่งง่าย และเป็นมิตรต่อผู้พัฒนามากกว่า PHP JavaScript ของ Codeigniter อย่างแน่นอน...

**ข้อ 9**

**ตอบ** 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 ได้อีกด้วย ซึ่งมีประโยชน์สารพัด ทำให้เราสามารถทดสอบ Web Application ที่เราพัฒนา บน Heroku ได้ฟรี เสมือนเรานำ Web Application มาทำการทดสอบบน Server จริงๆนั่นเอง

**ข้อ 10**

**ตอบ** วิชานี้ เป็นวิชาที่สำคัญอย่างมาก เพราะสาขาวิชาวิทยาการสารสนเทศนี้ มีหลักสูตรในการพัฒนาซอฟแวร์อยู่แล้ว เมื่อมีวิชานี้เข้ามา ทำให้เราสามารถพัฒนาซอฟแวร์ได้อย่างเป็นระบบ เป็นระเบียบมากขึ้น ทำให้ง่ายต่อการพัฒนา และเพื่อจุดประสงค์หลักคือ ตรงตามความต้องการของผู้ใช้ นั่นเอง...