1.

ก่อนอื่นเราต้องมาทำความรู้จักกับ agile กับ waterfall กันก่อนว่ามันคืออะไรยังไง

Agile เป็นหลักการในการพัฒนา software แบบใหม่ที่เน้น

* Rapid and flexible response to change
* ทำให้การพัฒนาว่องไว
* มีการทำเรื่อยๆไม่ต้องหยุด แม้มีอะไรมากระทบก็ไม่เป็นไร
* เมื่อมีการเปลี่ยนแปลง เราสามารถรองรับความเปลี่ยนแปลงนั้นได้อย่างรวดเร็ว ไม่ตายตัว

Waterfall คือกระบวนการพัฒนาระบบซอฟต์แวร์ที่เป็นขั้นเป็นตอนในรูปลักษณะของชั้นน้ำตก หรือ ขั้นบันได นับจากบนสุดไปยังชั้นล่างสุดแต่ละชั้นจะกินระยะเวลานานมากเหมาะสำหรับระบบซอฟต์แวร์ที่มีความต้องการที่แน่นอนชัดเจน ไม่คลุมเคลือ เพราะแต่ละขั้นตอนจะไม่มีการวนกลับมาทำซ้ำอีกในขั้นที่สูงกว่า แต่บางกรณีที่แบบจำลองตัวนี้ถูกพัฒนาให้สามารถวนกลับมาทำซ้ำขั้นตอนข้างบนได้ชั้นนึง

จากโจทย์ผมคิดว่า หลักการทำงานของทะเง agile และ waterfall ต่างก็เป็นหลักการทำงานที่ดีด้วยกันทั้งสองอย่างเพียงแต่ ควรเลือกใช้หลักการที่เหมาะกับการ เช่น agile ป็นการทำงานแบบใหม่มีการทำงานที่เร็วและสามารถเปลี่ยนแปลงแก้ไขได้เรื่อยๆ แต่อาจจะขาดความแน่ชัดของการคือ ไม่เหมาะกับงานที่ต้องการความชัดเจนที่แน่นอนมากๆ ส่วน หลักการทำงานแบบ waterfall เป็นการทำงานที่ค่อนข้างใช้ระยะเวลานานในการทำงานแต่ละขั้น แต่แลกมาด้วยงานที่มีความละเอียดและข้อผิดพลาดน้อย ซึ้งจากที่พูดมาข้างต้น สรุปได้ว่าหลักการทำงานทั้งสองแบบเป็นหลักการทำงานที่ดีด้วยกันทั้งคู่ เพียงแต่เราควรจะเลือกใช้ให้เหมาะสมกับงานที่เราจะนำหลักการเหล่านี้มาใช้ซะมากกว่า

2.

จากโจทย์ มันอาจจะเป็นเรื่องจริงก็ได้เพราะในปัจจุบัน git หรือ github ได้มีการพัฒนาในเรื่องของ version control ไปมาก จนในปัจจุบันมีหลัก 3 แบบด้วยกัน

* Version Control Systems แบบ Local
* ระบบ Version Control Systems แบบรวมศูนย์
* ระบบ Version Control Systems แบบกระจายศูนย์

ซึ่งทั้งสามแบบนี้สามาถนำมาแทนที่ CVS หรือ SVN ได้เลยก็เป็น คำกล่าวจากโจทย์ก็อาจจะเป็นไปได้ แต่ก็ไม่แน่เพราะอาจจะมีคนที่ไม่ถนัดในการใช้ git ก็อาจจะใช้งานแบบ CVS SVN ก็เป็นได้

3.

* git status
* git add .
* git commit -m "initial commit"
* git push origin master

4.

เมื่อมีการแก้ไขไฟล์ ไฟล์เดียวกันอยู่ จะทำให้ merge แล้วไม่เกิด conflict

5. ผลลัพธ์ของ ('a'..'e').each {|ch| print ch}

= > abcde"a".."e"

6.

จากโจทย์ การทำ web application ทำให้เกิดความสะดวก ง่าย และเร็ว สำหรับผู้พัฒนา app และ ผู้ใช้เพียงแค่มี internet แล้วก็ download ตัว app มาใช้งานได้เลย หากมีการ update ทางผู้ผลิต ก็แก้ไขเพียง app หลักแล้วก็สง่ตัว update ออกมาใช้ฝั่งผู้ใช้ได้ทำการ update เท่านั้นเอง แต่ การทำซอฟแวร์ลงใส่ใน CD มันจะทำการ update ซอฟแวร์เหล่านั้นได้ช้ากว่าและเข้าถึงตัวซอฟแวร์นั้นยากกว่าในปัจจุบัน เพราะในปัจจุบัน เกือบทุกคน บนโลกรู้จักและใช้งาน internet

7.

MVC มีการทำงานแบ่งเป็น 3 ส่วน คือ M= modal , V = view, C = controllerโดยแต่ละส่วนจะทำหน้าที่แตกต่างกันคือ View จะแสดงหน้าเว็บให้ผู้ใช้เห็น Model จะทำหน้าที่เชื่อมต่อข้อมูลกับ database และ Contoller จะทำหน้าที่เป็นตัวเชื่อมระหว่าง View และ Model

จากโจทย์

User ใช้งานผ่านหน้า index แล้ว Controller สั่งการให้ model ไปเรียกข้อมูลจาก database ออกมาแสดงในหน้า view แล้ว ส่งกลับมาแสดงให้ user เห็น

8.

Framework ที่เคยใช้คือ Object oriented programming(OOP) คือการมองทุกอย่างให้เป็น object ซึ่งแบ่งได้เป็น

1. Object ที่เป็น รูปธรรม เช่น คน,ใบรายการสินค้า

2. Object ที่เป็น นามธรรม เช่น ความเป็นเจ้าของ,เที่ยวบิน เป็นต้น

แต่แบบ rails เป็น Web Framework มีลักษณะเป็น MVC (Model-View-Controller)

Rails ถูกออกแบบมาให้มีการใช้งานที่ง่ายและรวดเร็ว ลดปัญหางานทีี่ต้องทำซ้ำๆ ทำให้ได้ productivity

ที่สูงขึ้น สอดคล้องกับ Methodology แบบ Ajile

ข้อดี ของ OOP

1. การนำกลับมาใช้ใหม่ คือเขียนโปรแกรมครั้งเดียว และนำกลับมาใช้ในหลาย ๆ ส่วนที่มีการทำงานเหมือนกัน
2. การซ้อนข้อมูล โดยจะทำให้ข้อมูลมีกลไกในการเข้าถึง และการมองเห็นได้ในหลาย ๆ รูปแบบ
3. การมีได้หลายรูปแบบ กล่าวให้ง่าย ๆ เลยก็คือวัตถุชนิดเดียวกัน หรือต่างชนิดกันสามารถมีการเรียกใช้คำสั่งที่เหมือนกัน แต่การทำงานของคำสั่งแตกต่างกันได้ตามคุณสมบัติของวัตถุชิ้นนั้น ๆ
4. การสืบทอดคุณสมบัติต่าง ๆ

ข้อเสีย ของ OOP

1. ซับซ้อนกว่าแบบ เชิงโครงสร้าง

ข้อดี ของ rails

1. Code เขียนง่าย ไม่ซับซ้อน เมื่อเทียบกับ java
2. ถ้า Code สั้น ดังนั้นโอกาสในการเกิด Bug จึงน้อย
3. เป็น Open source
4. มี Library ให้นำไปใช้เป็นจำนวนมาก
5. ง่ายต่อการนำไปขยายและพัฒนาต่อ (extend)
6. เป็น pure Object Oriented

9.

Heroku เป็น Platform as a Service (Paas) ที่ให้เราใช้งานได้ฟรี โดยรองรับภาษาโปรแกรมที่หลากหลาย เช่น Ruby, PHP, Node.js, Python, Java, Clojure, Scala และยังสามารถสร้าง buildpack สำหรับภาษาอื่นๆได้

10.

1. เพื่อทำความรู้จักกับการพัฒนา Software
2. เพื่อให้รุ้ถึงรูปแบบการทำงานแบบต่างๆ
3. เพื่อฝึกการทำงานเป็นกลุ่มและใช้งาน framework ต่างๆ